

**POLITECNICO DI MILANO**

DEPARTMENT OF PHYSICS  
School of Industrial and Information Engineering



MASTER'S DEGREE IN ENGINEERING PHYSICS

**Müller-Stokes Calculus  
for X-Ray Crystal Optics**

Candidate:  
**Edoardo Cappelli**

Supervisor:  
**Prof. Giacomo Ghiringhelli**

Assistant Supervisor:  
**Dr. Manuel Sanchez del Rio**

**Academic Year 2016-2017**



# Sommario

La capacità di controllare con precisione lo stato di polarizzazione di un fascio X si sta rivelando sempre più importante nelle beamline di sincrotrone specializzate nello studio di materiali magnetici. L'uso di cristalli come phase plate è una delle migliori alternative per raggiungere lo scopo, in quanto essi offrono alte rese e la possibilità di variare rapidamente energia ed elicità.

Lo scopo di questo progetto di tesi era creare una libreria Python per la simulazione di effetti di birifrangenza diffrattiva in phase plate ricavati da cristalli perfetti. Questa libreria doveva poi essere integrata nell'ambiente grafico OASYS per simulazioni di ottica di raggi X tramite widget dedicati, mettendo in questo modo a disposizione dell'utente un'interfaccia semplice e la facoltà di utilizzare le nuove funzionalità assieme ad altri software nell'ambito di un più ampio esperimento virtuale.

Il risultato di questo progetto è la libreria `crystalpy`. Essa lavora nel formalismo di Müller-Stokes e usa la teoria dinamica della diffrazione come presentata da Zachariasen per calcolare come la diffrazione da un cristallo perfetto influenzi lo stato di polarizzazione di un fascio X. Il pacchetto `crystalpy` in OASYS è ora pienamente integrato con SHADOW e in futuro sostituirà e rinnoverà il codice sui cristalli in ShadowOui.



# Abstract

The ability to precisely control the polarization state of X radiation is becoming more and more important in synchrotron beamlines which specialize in the study of magnetic materials. The use of crystal phase plates is one of the best alternatives to achieve this goal as they offer high yields, energy-tunability, and fast reversal of helicity.

The aim of this thesis work was to create a Python library for the simulation of diffractive birefringence effects in perfect crystal phase plates. This library was then to be integrated into the OASYS graphical environment for x-ray optics simulation through dedicated widgets, thereby offering both a user-friendly interface and the possibility to work hand in hand with other software and include the new functionalities into a full virtual experiment.

The result of this work is the `crystalpy` library. It works in the Müller-Stokes formalism and uses Zachariasen theory of dynamical diffraction to compute how the diffraction from a crystal phase plate affects the polarization state of an X-ray beam. The `crystalpy` package in OASYS is now fully integrated with SHADOW and will in time replace and renew the crystal code in ShadowOui.



# Contents

<b>Introduction</b>	<b>xi</b>
<b>1 Polarization Optics in the Stokes-Müller Formalism</b>	<b>1</b>
1.1 Plane waves . . . . .	1
1.2 The polarization ellipse . . . . .	3
1.3 Stokes parameters . . . . .	4
1.4 Müller matrices . . . . .	6
<b>2 The Dynamical Theory of X-Ray Diffraction</b>	<b>11</b>
2.1 Perfect crystals . . . . .	11
2.2 The structure factor . . . . .	14
2.3 The fundamental equations . . . . .	16
2.4 The dispersion surfaces . . . . .	21
2.5 The Bragg & Laue geometries . . . . .	22
<b>3 The OASYS Graphical Environment</b>	<b>27</b>
3.1 Beamline simulation software . . . . .	28
3.2 Virtual experiments in OASYS . . . . .	29
<b>4 The crystalpy Library</b>	<b>33</b>
4.1 The util toolbox . . . . .	33
4.2 The diffraction toolbox . . . . .	34
4.3 The polarization toolbox . . . . .	38
4.4 The oasys-crystalpy graphical interface . . . . .	41
4.5 The oasys-crystalpy widgets . . . . .	42
4.6 Anatomy of an Orange widget . . . . .	43
<b>5 Results</b>	<b>47</b>
5.1 Testing against Hirano <i>et al.</i> . . . . .	47
5.2 Testing against SHADOW . . . . .	53
<b>Conclusions</b>	<b>57</b>
<b>Appendix A.</b>	
<b>Bending Magnet Radiation</b>	<b>59</b>



# List of Figures

2.1	The Laue model of diffraction . . . . .	12
2.2	The Bragg model of diffraction . . . . .	13
2.3	Scattering from a single atom . . . . .	15
2.4	Scattering from a crystal unit cell . . . . .	15
2.5	The Ewald construction . . . . .	17
2.6	$D_0$ , $D_H$ and $\beta_H$ . . . . .	20
2.7	The $\sigma$ and $\pi$ field components . . . . .	21
2.8	Dispersion surfaces: far-out view . . . . .	22
2.9	Dispersion surfaces: close-up view . . . . .	23
2.10	The Laue and Bragg geometries . . . . .	24
2.11	Examples of intensity and phase retardation profiles. . . . .	26
3.1	SHADOW and SRW GUIs . . . . .	28
3.2	The virtual experiment . . . . .	30
3.3	The OASYS workflow . . . . .	31
3.4	A ShadowOui widget . . . . .	32
4.1	Bragg and Laue diffraction setups . . . . .	36
4.2	The <code>Diffraction</code> class . . . . .	37
4.3	The <code>PerfectCrystalDiffraction</code> class . . . . .	38
4.4	Rotated crystal reference frame: side view . . . . .	40
4.5	Rotated crystal reference frame: front view . . . . .	40
4.6	The <code>PhotonBunch</code> widget . . . . .	42
4.7	OASYS workflow for a crystal phase plate . . . . .	43
4.8	UML class diagram for the <code>util</code> toolbox. . . . .	46
5.1	Ge 220 Bragg diffraction at 10 keV . . . . .	48
5.2	Si 220 Bragg transmission at 10 keV . . . . .	48
5.3	Diamond 220 Bragg transmission at 10 keV . . . . .	49
5.4	Phase wrapping and number of points . . . . .	49
5.5	Si 220 Laue diffraction at 10 keV . . . . .	50
5.6	Ge 220 Laue diffraction at 20 keV . . . . .	50
5.7	Si 220 Laue transmission at 10 keV . . . . .	51
5.8	Ge 220 Laue transmission at 20 keV . . . . .	51

5.9	Bragg-Laue optics to produce CPX . . . . .	52
5.10	Circular polarization degree for Ge 220 Bragg-Laue reflections	52
5.11	Bragg diffraction in SHADOW and <code>crystalpy</code> . . . . .	53
5.12	Stokes parameters calculated with <code>crystalpy</code> . . . . .	54
5.13	Stokes parameters calculated with SHADOW . . . . .	54
5.14	Bending magnets in SHADOW and <code>crystalpy</code> . . . . .	55
5.15	Stokes parameters from <code>crystalpy</code> 's bending magnet . . . . .	55
5.16	Stokes parameters from SHADOW's bending magnet . . . . .	56
A.1	Schematic representation of a bending magnet . . . . .	59
A.2	Spacial distribution of bending magnet emission . . . . .	60
A.3	Stokes parameters with the <code>BendingMagnet</code> widget . . . . .	61

# Introduction

The polarization properties of synchrotron radiation play an outstanding role in experimental techniques such as non-resonant magnetic X-ray scattering [1, 2] and resonant exchange scattering [3]. In particular, circularly polarized photons have a definite angular momentum and can couple with magnetic moments in materials. This makes them an important probe for studies on magnetic materials [4].

Synchrotron radiation has specific polarization properties depending on the type of source:

- the radiation from a bending magnet (BM) is linearly polarized if observed in the orbit plane and elliptically polarized elsewhere, with different helicities (i.e. the sense of the electric field rotation) above and below the plane [5].
- the radiation from an undulator is mainly linearly polarized parallel to the plane of the electron oscillation (normal to the magnetic field) in the central radiation cone. There is also elliptical polarization out of the orbit plane and *helical undulators* can produce circularly polarized beams [6].

Circularly/elliptically-polarized X-rays can be obtained essentially in three ways: (i) viewing the source above or below the electron orbit plane, (ii) using special insertion devices to produce elliptical polarization on-axis [7], and (iii) using a phase plate to convert linear polarization into elliptical [8]. This last option offers several advantages, such as the possibility of fast reversal of helicity [9], a polarization state independent from changes of the electron orbit, and less disturbances to the polarization from other optical components [4].

Crystal phase plates work thanks to a phenomenon called *diffractive birefringence*: when X-rays diffract off a perfect crystal close to the Bragg condition, dynamical theory predicts the introduction of a phase shift  $\Phi$  between the two orthogonal components of the electric field  $E_\sigma$  and  $E_\pi$ , leading to elliptical polarization [8].

Simulating the behavior of these crystal plates helps to choose the right crystal plate (type of crystal, thickness, cut et cetera) and the appropriate

setup geometry for a certain experiment. This allows one to control the degree of circular polarization of the resulting radiation and the stability of the polarization state, as well as to tune the energy without changing the polarization.

During my traineeship at ESRF in Grenoble I developed the `crystalpy` Python library to simulate the diffraction of x-rays from perfect crystal phase plates. It was built upon previous work done by Mark Glass at ESRF<sup>1</sup>. The simulation of crystal diffraction close to a Bragg peak is performed using the formulas derived by Zachariasen [10] and the polarization state is described in the Müller-Stokes formalism.

I also integrated `crystalpy` in the OASYS Graphical Environment<sup>2</sup> [11] through several widgets with the aim of simplifying the process of setting the parameters and observing the results. This also allows the library to be interfaced with SHADOW [12] and other tools for beamline simulation which are presently being added to OASYS.

In this thesis work I will describe both the theoretical concepts and formalism upon which the algorithms are based and their implementation in the library itself. I will illustrate the basic features of the OASYS environment and the structure of the widgets I created. I will also present the reader with some results of calculations performed with the library and confront them with analogous results from published literature.

The thesis contents are organized as follows:

- Chapter 1 reviews some basic concepts about polarization states and introduces Stokes parameters and Müller matrices as a way to deal with partially polarized light. A way to switch between Müller-Stokes and Jones formalisms is also presented.
- Chapter 2 deals with the dynamical theory of diffraction, starting from perfect crystals and deriving the results that were used in the library.
- Chapter 3 provides an overview on the world of beamline software, focusing on the OASYS project and its goal of achieving a full virtual experiment.
- Chapter 4 describes schematically how the `crystalpy` library works and its integration into OASYS, the structure of an Orange widget and the Orange workflow.
- Chapter 5 shows the results that can be obtained with `crystalpy` and compares them with other programs and scientific literature.

---

<sup>1</sup>The code by Mark Glass can be found at <https://github.com/mark-glass/Orange-Crystal>.

<sup>2</sup>The OrAnge SYnchrotron Suite is being developed by Manuel Sanchez del Rio at ESRF and Luca Rebuffi at ELETTRA.

# Chapter 1

## Polarization Optics in the Stokes-Müller Formalism

"Christian Huygens was the first to suggest that light was not a scalar quantity, based on his work on the propagation of light through crystals; it appeared that light had 'sides' in the words of Newton. This vectorial nature of light is called *polarization*."

- Dennis Goldstein

### 1.1 Plane waves

X-rays, like all electromagnetic waves, can be described mathematically as traveling wave solutions to the Maxwell equations. In Maxwell's theory, time-varying electric fields generate magnetic fields and vice versa, accounting for the existence of self-perpetuating oscillations able to transport energy from one point in space to another [13, p. 269].

The force exerted on a test charge by an array of  $n$  other charges is the vector sum of the force exerted by any one charge individually, this is called the *superposition principle*. As a result of this, Maxwell's equations are linear and, in the absence of sources, they form a set of 4 homogeneous partial differential equations:

$$\begin{cases} \nabla \cdot \mathbf{E} = 0; & \nabla \times \mathbf{E} + \frac{1}{c} \frac{\partial \mathbf{B}}{\partial t} = 0 \\ \nabla \cdot \mathbf{B} = 0; & \nabla \times \mathbf{B} - \frac{\mu\epsilon}{c} \frac{\partial \mathbf{E}}{\partial t} = 0 \end{cases} \quad (1.1)$$

By applying the curl operator to the time-dependent equations and using the well-known vector identity  $\nabla \times (\nabla \times \mathbf{F}) = \nabla(\nabla \cdot \mathbf{F}) - \nabla^2 \mathbf{F}$ , we see that:

$$\nabla(\nabla \cdot \mathbf{E}) - \nabla^2 \mathbf{E} + \frac{1}{c} \frac{\partial}{\partial t} \left( \frac{\mu\epsilon}{c} \frac{\partial \mathbf{E}}{\partial t} \right) = 0$$

$$\nabla^2 \mathbf{E} - \frac{\mu\epsilon}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2} = 0$$

This is called the *wave equation*. Analogous equations can be written for each scalar component of the electric and magnetic fields:

$$\nabla^2 u - \frac{1}{v^2} \frac{\partial^2 u}{\partial t^2} = 0 \quad (1.2)$$

where  $v = c/\sqrt{\mu\epsilon}$  is the speed of light in a medium characterized by a magnetic permittivity  $\mu$  and a dielectric constant  $\epsilon$ .

The set of all *harmonic plane waves* of the form:

$$u = e^{i\mathbf{k}\cdot\mathbf{x} - i\omega t} \quad \text{with} \quad k = \frac{\omega}{v} \quad (1.3)$$

constitutes a vector space of fundamental solutions to the wave equation. They owe their name to the fact that they are constant over each of the planes

$$\mathbf{k} \cdot \mathbf{x} = \text{constant}$$

and they are harmonic with respect to time [14, pp. 14,16]. Simple plane wave solutions can then be linearly superimposed according to *Fourier integral theorem* to form a general solution [13, p. 68]. Considering the vector nature of the electric and magnetic fields and with the convention that the physical quantities are obtained by taking the real parts, we can write plane-wave fields of the form:

$$\begin{aligned} \mathbf{E}(\mathbf{x}, t) &= \mathbf{E}_0 e^{i\mathbf{k}\cdot\mathbf{x} - i\omega t} \\ \mathbf{B}(\mathbf{x}, t) &= \mathbf{B}_0 e^{i\mathbf{k}\cdot\mathbf{x} - i\omega t} \end{aligned} \quad (1.4)$$

with  $\mathbf{E}_0$ ,  $\mathbf{B}_0$  and  $\mathbf{k}$  constant in time and space. The divergence equations in (1.1) lead to:

$$\begin{aligned} \nabla \cdot (\mathbf{E}_0 e^{i\mathbf{k}\cdot\mathbf{x}}) e^{-i\omega t} &= 0 \\ \mathbf{E}_0 \cdot \mathbf{k} &= 0 \end{aligned} \quad (1.5)$$

and the same holds for the  $\mathbf{B}$  field. Thus, the oscillations of the fields take place on a plane perpendicular to the wave vector. Waves with such behavior are called *transverse waves* [13, p. 271]. Moreover, one can see that the curl equations imply that electric and magnetic fields be linked as follows:

$$\nabla \times \mathbf{E} + \frac{1}{c} \frac{\partial \mathbf{B}}{\partial t} = 0 \quad \Rightarrow \quad \mathbf{k} \times \mathbf{E}_0 = \frac{\omega}{c} \mathbf{B}_0 \quad (1.6)$$

If  $\mathbf{k}$  is real, this means  $\mathbf{E}_0$  and  $\mathbf{B}_0$  are *in phase* and perpendicular to each other.

Following this analysis we can say that, once a wave vector has been specified, a plane-wave solution can be fully determined by a two-dimensional

vector on the  $(\mathbf{E}, \mathbf{B})$  plane. This vector is in general complex and is called *Jones vector*:

$$\mathbf{j} = \begin{pmatrix} E_x \\ E_y \end{pmatrix} = \begin{pmatrix} E_{0,x} e^{i\delta_x} \\ E_{0,y} e^{i\delta_y} \end{pmatrix} \quad (1.7)$$

## 1.2 The polarization ellipse

At a point in space, the end point of the electric field describes a curve over time. The same curve would be obtained by looking at the projection of the end points onto a plane perpendicular to the direction of propagation at a fixed time.

If we define the  $z$  direction to be parallel to  $\mathbf{k}$  and we take the real part, the electric field can be written as:

$$\begin{cases} E_x = E_{0,x} \cos(\tau + \delta_x) \\ E_y = E_{0,y} \cos(\tau + \delta_y) \\ E_z = 0 \end{cases} \quad \text{where } \tau = \mathbf{k} \cdot \mathbf{x} - \omega t \quad (1.8)$$

We can find the curve's analytic expression by eliminating  $\tau$  from the previous equations as described in Born and Wolf [14, p. 25]. To do it, we start by expanding the cosine term according to  $\cos(\alpha + \beta) = \cos\alpha \cos\beta - \sin\alpha \sin\beta$ :

$$\begin{cases} \frac{E_x}{E_{0,x}} = \cos\tau \cos\delta_x - \sin\tau \sin\delta_x \\ \frac{E_y}{E_{0,y}} = \cos\tau \cos\delta_y - \sin\tau \sin\delta_y \end{cases}$$

therefore

$$\begin{cases} \frac{E_x}{E_{0,x}} \sin\delta_y - \frac{E_y}{E_{0,y}} \sin\delta_x = \cos\tau \sin\delta \\ \frac{E_x}{E_{0,x}} \cos\delta_y - \frac{E_y}{E_{0,y}} \cos\delta_x = \sin\tau \sin\delta \end{cases} \quad \text{where } \delta = \delta_y - \delta_x.$$

Squaring and adding both equations gives

$$\left(\frac{E_x}{E_{0,x}}\right)^2 + \left(\frac{E_y}{E_{0,y}}\right)^2 - 2\frac{E_x}{E_{0,x}}\frac{E_y}{E_{0,y}}\cos\delta = \sin^2\delta \quad (1.9)$$

This curve can be shown to be an ellipse and is called the *polarization ellipse*. It is independent of space and time coordinates and is thus an intrinsic characteristic of the plane wave. Different ellipses correspond to different *polarization states*. Some of the most interesting kinds of polarization correspond to degenerate forms of the ellipse: if the phase difference  $\delta$  is zero, the wave

is said to be *linearly polarized*, forming an angle  $\theta = \tan^{-1}(E_y/E_x)$  with the x direction; whereas if  $\delta = 90^\circ$  and  $E_x = E_y$  the wave is *circularly polarized*.

To sum up, a *polarization state* for a plane wave is characterized by a pair of components  $(E_{0,x}, E_{0,y})$  and a *phase difference*  $\delta$ . To each state we can associate an ellipse. This relation is not one-to-one though, because the electric field vector can trace an ellipse in two senses. For an observer looking at an oncoming optical beam, the polarization is *right-handed* if the tip of the electric field vector describes the ellipse in the clockwise sense [15, p. 58].

If we try to apply what we have derived until now to practical situations, we are faced with two main issues:

- Since the electric field vector traces the ellipse in roughly  $10^{-18}$  seconds for  $1\text{keV}$  x-rays, one cannot measure the polarization by following its rotation. It is then necessary to find some *measurable* parameters which can be associated to each polarization state.
- Strictly monochromatic light can always be associated to a well-defined polarization state, and is thus said to be *polarized*. Real light sources, however, never produce perfectly monochromatic light; in general, the variation of the field vectors is neither completely regular, nor completely irregular and the light is called *partially polarized*. In this case the two orthogonal components of the electric field do not have a constant phase difference and the observable effects depend on the correlation between them [14, p. 544].

We have to find an extended framework that we can apply to partially polarized light.

### 1.3 Stokes parameters

In 1852, Sir George Gabriel Stokes (1819–1903) discovered that the polarization behavior could be represented in terms of observables. He found that any state of polarized light could be completely described by four measurable quantities now known as the *Stokes polarization parameters* [15, p. 59].

Let's start by considering the case of a monochromatic plane wave as we did before. This wave will have a well-defined polarization state to which we can associate a polarization ellipse. We can only hope to measure the time-averaged behavior of the electric field vector, so we take the average of the fields along the ellipse:

$$\frac{\langle E_x^2(t) \rangle}{E_{0,x}^2} + \frac{\langle E_y^2(t) \rangle}{E_{0,y}^2} - 2 \frac{\langle E_x(t)E_y(t) \rangle}{E_{0,x}E_{0,y}} \cos \delta = \sin^2 \delta$$

where

$$\langle E_i(t)E_j(t) \rangle = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T E_i(t)E_j(t)dt \quad i, j = x, y.$$

After some calculations (see Goldstein [15, p. 61]) we get:

$$(E_{0,x}^2 + E_{0,y}^2) - (E_{0,x}^2 - E_{0,y}^2) - (2E_{0,x}E_{0,y} \cos \delta)^2 = (2E_{0,x}E_{0,y} \sin \delta)^2 \quad (1.10)$$

and we define the 4 *Stokes parameters* to be:

$$\left. \begin{aligned} s_0 &= E_{0,x}^2 + E_{0,y}^2 &= E_x E_x^* + E_y E_y^*, \\ s_1 &= E_{0,x}^2 - E_{0,y}^2 &= E_x E_x^* - E_y E_y^*, \\ s_2 &= 2E_{0,x}E_{0,y} \cos \delta &= E_x E_y^* + E_y E_x^*, \\ s_3 &= 2E_{0,x}E_{0,y} \sin \delta &= i(E_x E_y^* - E_y E_x^*); \end{aligned} \right\} \quad (1.11)$$

where  $E_\eta = E_{0,\eta} e^{i\delta_\eta} e^{i\mathbf{k}\cdot\mathbf{x} - i\omega t}$  and  $\eta = x, y$ .

It is common to group these parameters together in a *Stokes vector*, defined as

$$\mathbf{s} = \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} \quad (1.12)$$

We can now write equation (1.10) as

$$s_0^2 = s_1^2 + s_2^2 + s_3^2. \quad (1.13)$$

In Born and Wolf, Chapter 10.8, the "*polarization properties of quasi-monochromatic light*" are explored in detail, yielding the following:

$$\left. \begin{aligned} s_0 &= I(0^\circ, 0) + I(90^\circ, 0), \\ s_1 &= I(0^\circ, 0) - I(90^\circ, 0), \\ s_2 &= I(45^\circ, 0) - I(135^\circ, 0), \\ s_3 &= I\left(45^\circ, \frac{\pi}{2}\right) - I\left(135^\circ, \frac{\pi}{2}\right); \end{aligned} \right\} \quad (1.14)$$

where  $I(\theta, \epsilon)$  denotes the intensity of light vibrations in the direction making an angle  $\theta$  with the x-axis when a retardation  $\epsilon$  has been applied to the  $E_y$  component of the field.

These relations show how Stokes parameters can be calculated starting from measurable quantities (i.e. intensities) and bring into focus the connections between each parameter and a special kind of polarization state:  $s_0$  represents the *total intensity*,  $s_1$  is equal to the excess of intensity transmitted by a device accepting radiation along the  $\theta = 0^\circ$  direction over the intensity transmitted by a similar device accepting light along the  $\theta = 90^\circ$  direction;  $s_2$  and  $s_3$  have analogous interpretations, respectively with  $\theta = 45^\circ$  direction over  $\theta = 135^\circ$  direction and right-handed circular over left-handed circular.

If we now have partially polarized light, then equations 1.11 are valid for short time intervals, since amplitudes and phases fluctuate slowly [15, p. 62]. Using Schwarz's inequality, one can show that the Stokes parameters always satisfy this relation:

$$s_0^2 \geq s_1^2 + s_2^2 + s_3^2.$$

This is an equality for fully polarized light and an inequality for partially polarized light.

It is also possible to determine the *degree of polarization*, i.e. the ratio of the intensity that can be associated to specific polarization states and the total intensity:

$$P = \frac{I_{pol}}{I_{tot}} = \frac{\sqrt{s_1^2 + s_2^2 + s_3^2}}{s_0} \quad \text{with} \quad 0 \leq P \leq 1. \quad (1.15)$$

Furthermore, Stokes parameters are additive if we are dealing with the *incoherent superposition* of two or more beams (for a proof of this statement, see Goldstein [15, Chapter 5.6]). Using this property and acknowledging that  $\mathbf{s} = I_0 (1 \ 0 \ 0 \ 0)$  describes unpolarized light, we can write partially-polarized light through the following decomposition:

$$\mathbf{s} = \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = (1 - P)I_0 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + PI_0 \begin{pmatrix} 1 \\ s_1/PI_0 \\ s_2/PI_0 \\ s_3/PI_0 \end{pmatrix} \quad (1.16)$$

POLARIZATION STATE	STOKES VECTOR	$\delta$
Linear Horizontally Polarized (LHP)	$E_{0,x}^2 (1 \ 1 \ 0 \ 0)$	N/A
Linear Vertically Polarized (LVP)	$E_{0,y}^2 (1 \ -1 \ 0 \ 0)$	N/A
Linear $+45^\circ$ Polarized (L+45)	$2E_0^2 (1 \ 0 \ 1 \ 0)$	$0^\circ$
Linear $-45^\circ$ Polarized (L-45)	$2E_0^2 (1 \ 0 \ -1 \ 0)$	$180^\circ$
Right Circularly Polarized (RCP)	$2E_0^2 (1 \ 0 \ 0 \ 1)$	$90^\circ$
Left Circularly Polarized (LCP)	$2E_0^2 (1 \ 0 \ 0 \ -1)$	$270^\circ$

Table 1.1: Stokes vectors and phase differences for the most commonly encountered polarization states.

## 1.4 Müller matrices

Up to now we have stated what we mean when we talk about polarization states, polarized, unpolarized and partially-polarized light and we have built

a mathematical formulation that can be applied to all these concepts in the form of the Stokes parameters. The next step is to develop a formalism to describe the change in polarization properties when an electromagnetic wave interacts with matter.

Let us briefly list the different ways in which a polarization state can be changed and the names given to the corresponding optical devices:

- A *polarizer* or *diattenuator* changes the orthogonal amplitudes of the fields unequally;
- A *retarder* or *phase plate* changes the relative phase between orthogonal components of the fields;
- A *rotator* changes the direction of the electric field vector;
- A *depolarizer* transfers energy from the polarized states to the unpolarized state.

If we assume that the  $\{s'_i\}$  of the outgoing beam can be expressed as a linear combination of the 4 Stokes parameters of the incoming beam, we can define a matrix  $\mathcal{M}$  such that

$$\mathbf{s}' = \mathcal{M} \cdot \mathbf{s}; \quad (1.17)$$

such a matrix is called a *Müller matrix*.

Imagine we have an optical element of which we can model the optical properties so that we are able to express them in the form:

$$\mathbf{j}' = \mathcal{J} \cdot \mathbf{j} = \begin{pmatrix} \mathcal{J}_{00} & \mathcal{J}_{01} \\ \mathcal{J}_{10} & \mathcal{J}_{11} \end{pmatrix} \mathbf{j}, \quad \text{with } \{\mathcal{J}_{mn}\} \text{ complex}; \quad (1.18)$$

here  $\mathbf{j}$  is a Jones vector as defined in (1.7) and  $\mathcal{J}$  is the *Jones matrix* that converts one electric field configuration to another. For each Jones matrix one can always find a Müller matrix which represents the same optical element in the Stokes parameters formalism (i.e. Müller-Stokes formalism); the inverse is not true, e.g. the Müller matrix for a depolarizer cannot have a matching Jones matrix, since the latter is defined only for fully-polarized plane waves [15, p. 150]. To find out how we can transform one into the other, let us follow the derivation found in [16, Appendix 4].

The *Kronecker product* between an  $m \times n$  matrix  $\mathcal{A}$  and a  $p \times q$  matrix  $\mathcal{B}$  is defined as the  $mp \times nq$  block matrix

$$\mathcal{A} \otimes \mathcal{B} = \begin{pmatrix} a_{00}\mathcal{B} & \dots & a_{0n}\mathcal{B} \\ \vdots & \ddots & \vdots \\ a_{m0}\mathcal{B} & \dots & a_{mn}\mathcal{B} \end{pmatrix} \quad (1.19)$$

and has the *mixed-product property*:

$$(\mathcal{A}\mathcal{C}) \otimes (\mathcal{B}\mathcal{D}) = (\mathcal{A} \otimes \mathcal{B})(\mathcal{C} \otimes \mathcal{D}). \quad (1.20)$$

From (1.19)

$$\mathbf{j} \otimes \mathbf{j}^* = \begin{pmatrix} E_x \\ E_y \end{pmatrix} \otimes \begin{pmatrix} E_x^* \\ E_y^* \end{pmatrix} = \begin{pmatrix} E_x E_x^* \\ E_x E_y^* \\ E_y E_x^* \\ E_y E_y^* \end{pmatrix}.$$

From (1.11)

$$\mathbf{s} = \begin{pmatrix} E_x E_x^* + E_y E_y^* \\ E_x E_x^* - E_y E_y^* \\ E_x E_y^* + E_y E_x^* \\ i(E_x E_y^* + E_y E_x^*) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \\ 0 & i & -i & 0 \end{pmatrix} (\mathbf{j} \otimes \mathbf{j}^*) \equiv \mathcal{H}(\mathbf{j} \otimes \mathbf{j}^*)$$

From (1.18) and (1.20)

$$\mathbf{s}' = \mathcal{H}(\mathbf{j}' \otimes \mathbf{j}'^*) = \mathcal{H}(\mathcal{J}\mathbf{j} \otimes \mathcal{J}^*\mathbf{j}^*) = \mathcal{H}(\mathcal{J} \otimes \mathcal{J}^*)\mathcal{H}^{-1}\mathcal{H}(\mathbf{j} \otimes \mathbf{j}^*).$$

Thus, according to (1.17),

$$\boxed{\mathcal{M} = \mathcal{H}(\mathcal{J} \otimes \mathcal{J}^*)\mathcal{H}^{-1}}. \quad (1.21)$$

With this equation we are able to calculate the Müller matrix for any optical object provided we can find a linear relation between the Jones vectors of the incoming and outgoing fields.

The matrix elements can also be found by means of the *Pauli matrices* as derived by Huard [17] using the following formula:

$$\mathcal{M}_{ij} = \frac{1}{2} \text{Tr} \left( \mathcal{J} \sigma_j \mathcal{J}^\dagger \sigma_i \right)$$

where  $\dagger$  stands for *Hermitian transpose* and the Pauli matrices are:

$$\sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \quad (1.22)$$

ELEMENT	MODEL	MÜLLER
Polarizer	$E'_x = p_x E_x$ $E'_y = p_y E_y$	$\begin{pmatrix} (p_x^2 + p_y^2)/2 & (p_x^2 - p_y^2)/2 & 0 & 0 \\ (p_x^2 - p_y^2)/2 & (p_x^2 + p_y^2)/2 & 0 & 0 \\ 0 & 0 & p_x p_y & 0 \\ 0 & 0 & 0 & p_x p_y \end{pmatrix}$
Retarder	$E'_x = e^{i\phi/2} E_x$ $E'_y = e^{-i\phi/2} E_y$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos \phi & \sin \phi \\ 0 & 0 & -\sin \phi & \cos \phi \end{pmatrix}$
Rotator	$E'_x = E_x \cos \theta + E_y \sin \theta$ $E'_y = -E_x \sin \theta + E_y \cos \theta$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 2\theta & \sin 2\theta & 0 \\ 0 & -\sin 2\theta & \cos 2\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Table 1.2: Müller matrices for the three fundamental polarization elements: polarizer ( $\Delta\text{phase} = \phi$ ) and rotator ( $\Delta\text{angle} = \theta$ ).



## Chapter 2

# The Dynamical Theory of X-Ray Diffraction from Perfect Crystals

"The environment of a particular atom in a crystal has a certain arrangement, and if you look at the same kind of an atom at another place farther along, you will find one whose surroundings are exactly the same. If you pick an atom farther along by the same distance, you will find the conditions exactly the same once more. The pattern is repeated over and over again — and, of course, in three dimensions".

- *Richard Feynman*

### 2.1 Perfect crystals

In the following few pages we are going to derive the basic equations governing the diffraction of x-rays from perfect crystals according to *dynamical theory*, roughly following the treatments by Zachariasen and Batterman and Cole. As a first step, it is necessary to define some concepts from the physics of perfect crystals.

#### A. The lattice abstraction

A *perfect crystal* is an idealized model of a real crystal, consisting of a three-dimensional array of repeating "blocks", filling space without leaving empty portions. These "blocks" can be single atoms, groups of atoms, molecules or ions and are called *unit cells*.

A *Bravais lattice* is an infinite array of discrete points with position vectors  $\mathbf{R}$  of the form

$$\mathbf{R} = n_1\mathbf{a}_1 + n_2\mathbf{a}_2 + n_3\mathbf{a}_3 \quad (2.1)$$

where the *primitive vectors*  $\mathbf{a}_{1,2,3}$  span 3D lattice and  $n_{1,2,3}$  range through all integers [18, pp. 64,65].

For each crystal Bravais lattice (also called *real lattice*), one can define a *reciprocal lattice* as the set of all wave vectors  $\mathbf{H}$  yielding plane waves with the same periodicity as the crystal lattice itself. It can be proved that the reciprocal lattice is a Bravais lattice in its own right, with primitive vectors

$$\mathbf{b}_1 = 2\pi \frac{\mathbf{a}_2 \times \mathbf{a}_3}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)}, \quad \mathbf{b}_2 = 2\pi \frac{\mathbf{a}_3 \times \mathbf{a}_1}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)}, \quad \mathbf{b}_3 = 2\pi \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)}$$

and lattice vectors

$$\mathbf{H} = h_1 \mathbf{b}_1 + h_2 \mathbf{b}_2 + h_3 \mathbf{b}_3 \quad \text{where the } \{h_i\} \text{ are integers.} \quad (2.2)$$

It is easy to see how the  $\{b_i\}$  we chose respect the definition of reciprocal lattice. In fact:

$$\mathbf{b}_i \cdot \mathbf{a}_j = 2\pi \delta_{ij}$$

and

$$\mathbf{H} \cdot \mathbf{R} = 2\pi(h_1 n_1 + h_2 n_2 + h_3 n_3);$$

thus

$$e^{i\mathbf{H} \cdot (\mathbf{x} + \mathbf{R})} = e^{i\mathbf{H} \cdot \mathbf{x}} e^{i\mathbf{H} \cdot \mathbf{R}} = e^{i\mathbf{H} \cdot \mathbf{x}}. \quad (2.3)$$

A *lattice plane* is a plane containing at least three non-collinear Bravais lattice points and a *family of lattice planes* is a set of equidistant parallel lattice planes which together contain every point of the Bravais lattice.

According to the previous definitions, one can prove that to each family of lattice planes separated by a distance  $d$  corresponds a set of reciprocal lattice vectors orthogonal to the planes, the shortest of which has modulus  $2\pi/d$  [18, p. 90]. This shortest reciprocal vector can be used to unambiguously identify a family of lattice planes through its components with respect to a set of basis vectors. A lattice plane has *Miller indices*  $h, k, l$  if the shortest reciprocal lattice vector orthogonal to it can be written as  $\mathbf{H} = h\mathbf{b}_1 + k\mathbf{b}_2 + l\mathbf{b}_3$ .

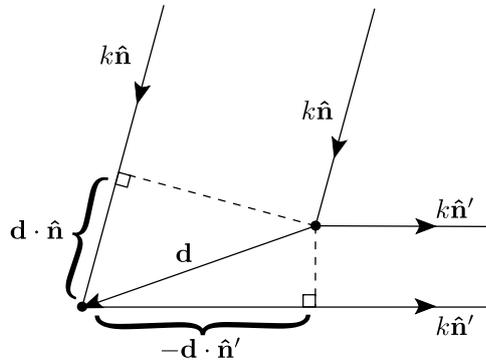


Figure 2.1: A schematic depiction of the elastic scattering of radiation from two lattice points separated by a distance vector  $\mathbf{d}$ .  $\mathbf{k}$  and  $\mathbf{k}'$  are respectively the wavevectors of the incident and the scattered radiation, their moduli are  $k = k' = 2\pi/\lambda$ .

## B. The Laue and Bragg equations

The von Laue approach to crystal diffraction consists in considering the lattice as made up of individual scatterers (sets of atoms or ions) situated at the Bravais lattice points. Intensity peaks will occur if the elastically scattered beams from these points interfere constructively. This happens when the optical path difference is an integral number of wavelengths. Following the simple geometrical model displayed in figure 2.1 for two scatterers, we get:

$$\mathbf{d} \cdot \mathbf{n} - \mathbf{d} \cdot \mathbf{n}' = m\lambda \implies \mathbf{d} \cdot (\mathbf{k} - \mathbf{k}') = 2\pi m \quad (2.4)$$

We can extend this condition to the whole lattice by writing

$$\mathbf{R} \cdot (\mathbf{k} - \mathbf{k}') = 2\pi m \implies e^{\mathbf{R} \cdot (\mathbf{k} - \mathbf{k}')} = 1 \quad \forall \mathbf{R} \in \text{Bravais lattice.}$$

This is the *Laue condition*: constructive interference will occur if and only if  $\mathbf{H} = \mathbf{k} - \mathbf{k}'$  is a reciprocal lattice vector. Thus, to each reciprocal lattice vector corresponds one diffraction peak with wavevector given by the *Laue equation*:

$$\boxed{\mathbf{k}^H = \mathbf{k}^0 + \mathbf{H}} \quad (2.5)$$

Furthermore, in elastic scattering  $|\mathbf{k}^H| = |\mathbf{k}^0| = k$  and, as we said before,

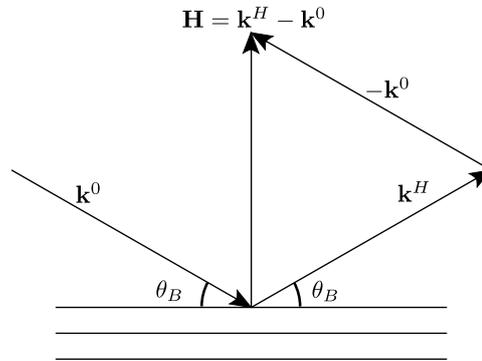


Figure 2.2: First order of diffraction from the family of lattice planes associated to the reciprocal vector  $\mathbf{H}$ .  $\theta_B$  is the *Bragg angle*.

the reciprocal lattice vectors corresponding to a family of lattice planes are orthogonal to the planes themselves and have moduli  $k = (2\pi/d)m$ . As we can see in figure 2.2, by simply drawing a sketch of the system we can appreciate how the vectors  $\mathbf{k}^0$  and  $\mathbf{k}^H$  make equal angles with the lattice planes. We call this angle the *Bragg angle*,  $\theta_B$ . We can now easily derive the

famous *Bragg equation* just from these geometrical considerations:

$$\left. \begin{aligned} H &= \frac{2\pi}{d}m \\ H &= 2k \sin \theta_B \\ k &= \frac{2\pi}{\lambda} \end{aligned} \right\} \implies \boxed{2d \sin \theta_B = m\lambda}. \quad (2.6)$$

Inside a crystal, if surface effects can be neglected and the infinite crystal abstraction can be applied, all physical properties are going to have the translational symmetry of the Bravais lattice; meaning that a property  $\Omega(\mathbf{x}) = \Omega(\mathbf{x} + \mathbf{R})$  can be expanded in a *Fourier series* as:

$$\Omega(\mathbf{x}) = \sum_{H_1} \sum_{H_2} \sum_{H_3} \Omega_{H_1 H_2 H_3} e^{i(H_1 x_1 + H_2 x_2 + H_3 x_3)} = \sum_{\mathbf{H}} \Omega_{\mathbf{H}} e^{i\mathbf{H} \cdot \mathbf{x}} \quad (2.7)$$

with

$$\Omega_{\mathbf{H}} = \frac{1}{V} \int_V \Omega(\mathbf{x}) e^{-i\mathbf{H} \cdot \mathbf{x}} d\mathbf{v} \quad (2.8)$$

where  $V$  is the volume of the unit cell (the spacial period of the function) and the sum is over all the reciprocal lattice vectors, i.e. over all the plane waves which have the periodicity of the lattice (as expressed in equation 2.3).

## 2.2 The structure factor

All diffraction phenomena happening in a crystal can ultimately be traced back to the interaction between electrons and electromagnetic waves. We will limit our treatment to a classical description.

Let us consider an atom with  $z$  electrons orbiting a nucleus which stays still at the origin of the coordinate system. In fact, any motion of the nucleus (e.g. thermal vibrations) will take place on a much longer time frame, as it is thousands of times more massive than an electron, and the dynamics of the electron cloud can be effectively decoupled from that of the nuclei (*Born-Oppenheimer approximation*). Each electron will be accelerated by the incoming wave field and will in turn radiate as an oscillating electric dipole [13, p. 658], (14.18).

If the distances  $\{\mathbf{x}_j\}$  between the electrons and the nucleus are much smaller than the distances  $\{\mathbf{R}_j\}$  to the observation point, we can consider the emitted radiation field as a linear superposition of plane waves. With this approximation we can express the total emitted radiation as the product between the free-electron emission and a phase term:

$$\mathbf{E}_{atom} = \mathbf{E}_e \sum_j e^{-i(\mathbf{k}-\mathbf{k}^0) \cdot \mathbf{x}_j} = \mathbf{E}_e \sum_j e^{i\mathbf{s} \cdot \mathbf{x}_j}. \quad (2.9)$$

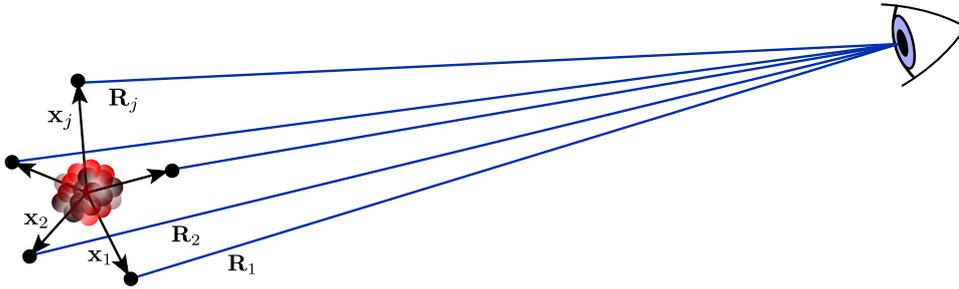


Figure 2.3: Schematic classical representation of the atomic system at a given instant. The proportions are greatly exaggerated.

The  $\{\mathbf{s} \cdot \mathbf{x}_j\}$  represent the phase difference between the radiation emitted by an electron at  $\mathbf{x}_j$  and an electron at the origin; X-ray wavelengths are comparable to these distances, so these terms cannot be neglected. Since the electrons move very rapidly, however, in the time scales we are considering the radiation is effectively averaged over an electron "orbit" [10, p. 92]. To take into account only the average positions for each electron, we can substitute the electrons for a charge distribution function  $\sigma(\mathbf{x})$ , with  $\sigma(\mathbf{x})d\mathbf{v}$  being the electric charge in volume  $d\mathbf{v}$ . The mean amplitude then becomes

$$\bar{\mathbf{E}}_{atom} = \mathbf{E}_e \underbrace{\int \sigma(\mathbf{x}) e^{i\mathbf{s} \cdot \mathbf{x}} d\mathbf{v}}_{\equiv f(\mathbf{s})}. \quad (2.10)$$

where  $f(\mathbf{s})$  is the *atomic form factor* [19, p. 11].

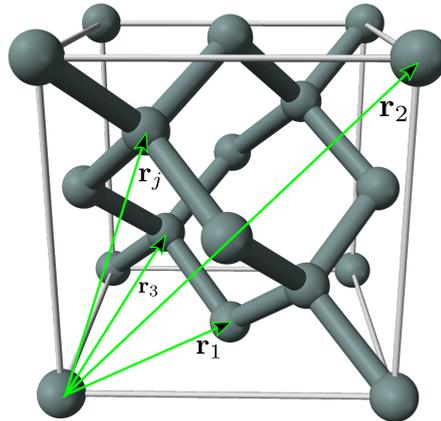


Figure 2.4: Face-centered cubic structure with a two-atom basis (e.g. silicon).

We now take this reasoning one step further by considering the scattering from a single unit cell of the crystal lattice: we fix the origin of the coordinates at a corner of the unit cell and label the position of the  $k$ -th atom in the cell with a vector  $\mathbf{r}_k$  as shown in figure 2.4. By following the same logic as before:

$$\mathbf{E}_{cell} = \mathbf{E}_e \int \Omega(\mathbf{x}) e^{i\mathbf{s}\cdot\mathbf{x}} \quad (2.11)$$

where  $\Omega(\mathbf{x})$  is the distribution of electric charge throughout the unit cell volume. This distribution is roughly equal to the sum of the atomic charge distributions  $\{\sigma_k\}$  for each atom in the unit cell. We can express the amplitude emitted by the unit cell as  $F \mathbf{E}_e$ , where  $F$  is called "scattering power of the unit cell" or, more commonly, *structure factor* [19, p. 151]:

$$F = \int \Omega(\mathbf{x}) e^{i\mathbf{s}\cdot\mathbf{x}} \approx \sum_k f_k(\mathbf{s}) e^{i\mathbf{s}\cdot\mathbf{r}_k} \quad (2.12)$$

### 2.3 The fundamental equations

Diffraction from a thick crystal is a phenomenon characterized by the interplay between incident and scattered fields: the scattered wave depends on the electric field felt by the electrons in the medium, and this is in turn shaped by both incident and scattered fields. This contribution of the scattered field is a second order effect and is of particular importance only when the crystal is nearly perfect and big enough to generate a significant diffracted amplitude. This is the domain of the dynamical theory of diffraction, which is thus based upon a self-consistent set of coupled fields inside the crystal, constrained by the boundary conditions at the surfaces<sup>1</sup>.

First, let us describe the orientation of the crystal in space by a unit normal  $\hat{\mathbf{n}}$  pointing *outwards* in the  $z$  direction and let us fix the origin of the coordinates on the surface plane, so that it can be described as  $\hat{\mathbf{n}} \cdot \mathbf{x} = 0$ . The *external* field  $\mathbf{E}_0^e$  and the *internal* field  $\mathbf{D}_0$  are:

$$\left. \begin{aligned} \mathbf{E}^e &= \mathbf{E}_0^e e^{i\mathbf{k}_0^e \cdot \mathbf{x} - i\omega t} \\ \mathbf{D} &= \mathbf{D}_0 e^{i\beta_0 \cdot \mathbf{x} - i\omega t} \end{aligned} \right\} \quad (2.13)$$

The *refractive index* in the x-ray region is only slightly different from unity, so we can write it as  $1 + \delta_0$  with  $\delta_0$  a very small quantity and find an

---

<sup>1</sup>We will restrict our treatment to the diffraction of monochromatic waves from perfect crystals. Crystal imperfections and/or spacial and spectral dispersion can be introduced through a posteriori corrections.

approximate relation between  $\beta_0$  and  $\mathbf{k}_0^e$ :

$$\left. \begin{aligned} \beta_0^2 &= k_0^2(1 + \delta_0)^2 \approx k_0^2(1 + 2\delta_0) \\ \beta_0 &= \mathbf{k}_0^e + \Delta\hat{\mathbf{n}} \end{aligned} \right\} \implies \beta_0 = \mathbf{k}_0^e + \frac{k_0\delta_0}{\gamma_0}\hat{\mathbf{n}}$$

at the boundary the exponential functions must agree,  
so the wave vectors can only differ by a multiple of the normal.

(2.14)

where  $\gamma_0 \equiv -\hat{\mathbf{n}} \cdot \mathbf{u}_0^e$ .

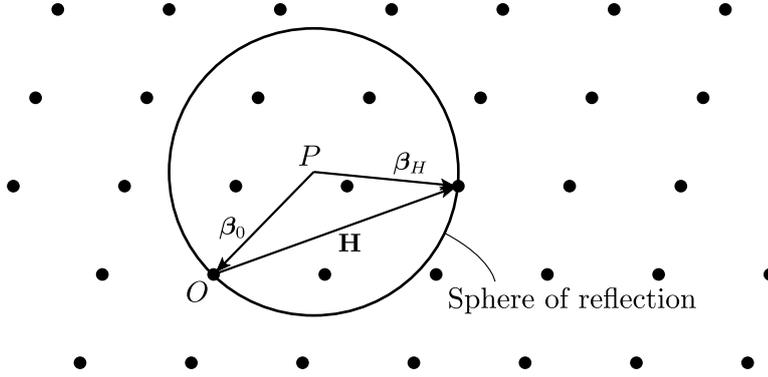


Figure 2.5: *The Ewald construction*:  $O$  is the origin of the reciprocal lattice,  $\beta_0$  is the wavevector of the incident radiation inside the crystal, to each lattice point corresponds a reciprocal vector. The Laue condition for diffraction is satisfied only if at least one reciprocal lattice point lies on the Ewald sphere (a circumference in this 2D representation), and in that case the vector connecting the centre of the sphere to that point is the diffracted wavevector.

While the dynamical theory is necessary to provide an accurate description of the intensities of the diffracted waves, the directions are well predicted by the kinematical approach: the amplitude of scattering is negligible unless the Laue vector equation is exactly or very nearly satisfied [10, p. 113]. The internal wave field can thus be written as the sum of the incident field and the diffracted fields, each one satisfying  $\beta_H = \beta_0 + \mathbf{H}$ :

$$\mathbf{D} = \sum_H \mathbf{D}_H e^{i\beta_H \cdot \mathbf{x} - i\omega t} \quad (2.15)$$

Referring to figure 2.5, we will have that  $\mathbf{D}_H \approx 0$  for each reciprocal lattice point  $\mathbf{H}$  lying relatively far from the sphere of reflection. Specifically, we shall henceforth assume that the crystal produces only one scattered beam at a time, i.e. only the wave fields  $\mathbf{D}_0$  and one  $\mathbf{D}_H$  have significant amplitudes (*two-wave approximation*).

### A. The dielectric constant of the crystal lattice

As we have discussed in the previous section, the crystal responds to an electric field by having its electrons displaced from their ground state positions and the material can be said to have a net instantaneous *polarization*  $\mathbf{p}e^{-i\omega t}$ . This polarization vector has very little to share with the *optical* polarization vector and one has to think of it as a "volume density of electric dipoles". The material's *polarizability*  $\alpha$  is defined so that:

$$\mathbf{p} = \alpha \mathbf{E}. \quad (2.16)$$

Remembering the distribution  $\Omega$  from (2.11), we have that the polarizability for the entire unit cell is

$$\alpha(\mathbf{x}) = -\frac{e^2}{m\omega^2}\Omega(\mathbf{x}). \quad (2.17)$$

We follow the notation in [10] and define  $\psi = 4\pi\alpha$ . The *dielectric constant* of the crystal is thus:

$$\epsilon \approx 1 + 4\pi\alpha = 1 + \psi. \quad (2.18)$$

The polarizability has the same periodicity as the crystal lattice, so we can expand it in a Fourier series as described in (2.7) and (2.8):

$$\psi = 4\pi\alpha = 4\pi \sum_H \alpha_H e^{i\mathbf{H}\cdot\mathbf{x}}, \quad \alpha_H = \frac{1}{V} \int_V \alpha e^{-i\mathbf{H}\cdot\mathbf{x}} dv \quad (2.19)$$

In the two-wave approximation the phase relation between incident and diffracted wave depends on the pair of wavevectors  $(\boldsymbol{\beta}_0, \boldsymbol{\beta}_H)$  satisfying the Laue condition for a specific vector  $\mathbf{H}$  of the reciprocal lattice. For a particular  $\mathbf{H}$ , we have a particular structure factor  $F_H$ , given by:

$$\begin{aligned} F_H &= \int \Omega(\mathbf{x}) e^{-i(\boldsymbol{\beta}_H - \boldsymbol{\beta}_0)\cdot\mathbf{x}} dv \\ &= \int \Omega(\mathbf{x}) e^{-i\mathbf{H}\cdot\mathbf{x}} dv \end{aligned} \quad (2.20)$$

Putting together (2.17), (2.19), and (2.20) we get:

$$\psi_H = 4\pi\alpha_H = -\frac{4\pi e^2 F_H}{m\omega^2 V} = -\frac{r_e \lambda^2 F_H}{V\pi} \quad (2.21)$$

where  $r_e = e^2/mc^2$  is the *classical electron radius* and  $V$  is the volume of the unit cell.

### B. The general case

Now that we have described the internal field as a sum of plane waves whose wavevectors satisfy the Laue condition (2.15) and we have calculated the crystal's dielectric constant (2.18), we can proceed to compute the intensities of the different wave components by applying Maxwell's equations in the crystal lattice. But first let us make some assumptions:

- $\psi$  is a small quantity at x-ray frequencies,  $\mathbf{E} \approx (1 - \psi)\mathbf{D}$ ;
- the magnetic permeability is negligible,  $\mathbf{B} \approx \mathbf{H}$ ;
- there are no conduction currents,  $\mathbf{J}_f \approx 0$ .

$$\left. \begin{aligned} \nabla \times \mathbf{H} &= \frac{1}{c} \frac{\partial \mathbf{D}}{\partial t} \\ \nabla \times \mathbf{E} &= -\frac{1}{c} \frac{\partial \mathbf{B}}{\partial t} \end{aligned} \right\} \implies \nabla \times (\nabla \times (1 - \psi)\mathbf{D}) = -\frac{1}{c^2} \frac{\partial^2 \mathbf{D}}{\partial t^2} \quad (2.22)$$

Let us focus on the left hand side of (2.22). Using (2.15) and (2.19), we get:

$$(1 - \psi)\mathbf{D} = e^{-i\omega t} \left\{ \sum_H \mathbf{D}_H e^{i\beta_H \cdot \mathbf{x}} - \sum_K \sum_L \psi_K \mathbf{D}_L e^{i(\beta_L + \mathbf{K}) \cdot \mathbf{x}} \right\} \quad (2.23)$$

where summing over  $H$ ,  $K$  or  $L$  means summing over the whole reciprocal lattice. From our definition of  $\beta_H$  as  $\beta_H = \beta_0 + \mathbf{H}$ , we can say that

$$\beta_L + \mathbf{K} = \beta_0 + \mathbf{L} + \mathbf{K} = \beta_{L+\mathbf{K}} \equiv \beta_H$$

and

$$\sum_K \sum_L \psi_K \mathbf{D}_L \equiv \sum_H \mathbf{C}_H.$$

Moreover, one can easily prove that  $\nabla \times (e^{\mathbf{a} \cdot \mathbf{x}} \mathbf{A}) = e^{\mathbf{a} \cdot \mathbf{x}} (\mathbf{a} \times \mathbf{A} + \nabla \times \mathbf{A})$ . In our case  $\psi_K$  and  $\mathbf{D}_L$  are not functions of the position, since they are Fourier coefficients, so  $\nabla \times \mathbf{C}_H = 0$ . By applying this formula to the previous equation:

$$\nabla \times (\nabla \times (1 - \psi)\mathbf{D}) = -e^{-i\omega t} \left\{ \sum_H \beta_H \times (\beta_H \times (\mathbf{D}_H - \mathbf{C}_H)) e^{i\beta_H \cdot \mathbf{x}} \right\}.$$

The right hand side of (2.22) can be written as:

$$-\frac{1}{c^2} \frac{\partial^2 \mathbf{D}}{\partial t^2} = e^{-i\omega t} \left\{ \frac{\omega^2}{c^2} \sum_H \mathbf{D}_H e^{i\beta_H \cdot \mathbf{x}} \right\}.$$

Equation (2.22) holds only if all the corresponding coefficients of the Fourier expansions are equal:

$$-\beta_H \times (\beta_H \times (\mathbf{D}_H - \mathbf{C}_H)) = \frac{\omega^2}{c^2} \mathbf{D}_H = k_0^2 \mathbf{D}_H \quad (2.24)$$

By applying the vector identity  $\mathbf{a} \times (\mathbf{a} \times \mathbf{b}) = (\mathbf{a} \cdot \mathbf{b})\mathbf{a} - a^2\mathbf{b}$  and by noticing that (2.24) implies  $\mathbf{D}_H \cdot \boldsymbol{\beta}_H = 0$ , this can be rewritten in the more intelligible form:

$$\boxed{\sum_L \left\{ \psi_{H-L} (\boldsymbol{\beta}_H \cdot \mathbf{D}_L) \boldsymbol{\beta}_H - \psi_{H-L} \beta_H^2 \mathbf{D}_L \right\}} = (k_0^2 - \beta_H^2) \mathbf{D}_H \quad (2.25)$$

This is the fundamental system of equations for the dynamical theory: in principle, every diffracted wave should be coupled to all the other waves which satisfy the Laue condition. In practice, as we have said previously, it is almost always the case that an incoming field only excites one diffracted wave at a time [10, p. 113]. In the following we will work in this approximation.

### C. The two-wave approximation

The system in (2.24) becomes a system of two equations:

$$\begin{cases} \psi_{\bar{H}} (\boldsymbol{\beta}_0 \cdot \mathbf{D}_H) \boldsymbol{\beta}_0 - \psi_{\bar{H}} \beta_0^2 \mathbf{D}_H &= (k_0^2 - \beta_0^2 (1 - \psi_0)) \mathbf{D}_0 \\ \psi_H (\boldsymbol{\beta}_H \cdot \mathbf{D}_0) \boldsymbol{\beta}_H - \psi_H \beta_H^2 \mathbf{D}_0 &= (k_0^2 - \beta_H^2 (1 - \psi_0)) \mathbf{D}_H \end{cases} \quad (2.26)$$

where  $\bar{H} \equiv -H$ .

The second equation implies that  $\mathbf{D}_H$  is linearly dependent on  $\boldsymbol{\beta}_H$  and  $\mathbf{D}_0$ , which means they all lie in the same plane and  $\mathbf{D}_H$  is normal to  $\boldsymbol{\beta}_H$  as we stated before. These vector equations can be turned into scalar by taking the scalar product respectively with  $\mathbf{D}_0$  and  $\mathbf{D}_H$ :

$$\begin{cases} (2\delta_0 - \psi_0) D_0 - \psi_{\bar{H}} \sin \chi D_H &= 0 \\ -\psi_H \sin \chi D_0 + (2\delta_H - \psi_0) D_H &= 0 \end{cases} \quad (2.27)$$

where  $\chi$  is the angle between  $\mathbf{D}_0$  and  $\boldsymbol{\beta}_H$  as shown in figure 2.6. The equa-

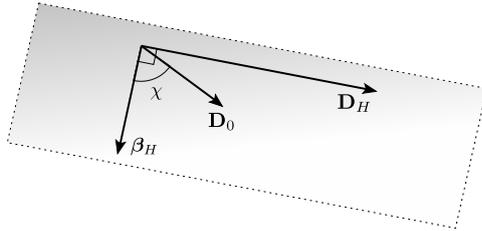


Figure 2.6: The geometry of the  $\mathbf{D}_0$ ,  $\mathbf{D}_H$  and  $\boldsymbol{\beta}_H$  vectors.

tions in (2.27) are the self-consistency conditions for incoming and diffracted fields and they have a non-trivial solution only if the determinant vanishes, i.e. if:

$$(2\delta_0 - \psi_0)(2\delta_H - \psi_0) = \psi_H \psi_{\bar{H}} \sin^2 \chi \quad (2.28)$$

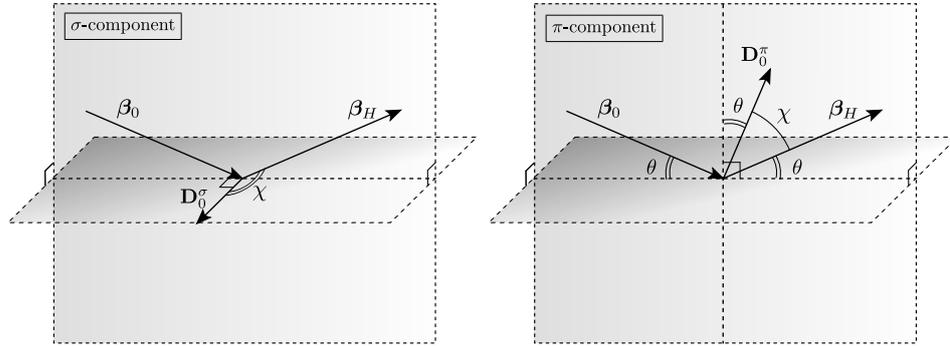
and the solution is:

$$x \equiv \frac{D_H}{D_0} = \frac{2\delta_0 - \psi_0}{\psi_{\bar{H}} \sin \chi}. \quad (2.29)$$

## 2.4 The dispersion surfaces

In the previous section we have found the general solution to the coupled system describing crystal diffraction in the two-wave approximation. Next we will develop a graphical representation of the scattering process which extends the simple Ewald construction seen in figure 2.5 to the more general case we are considering.

As a first step, let us describe the incident wave field as the sum of two orthogonal fields  $\sigma$  and  $\pi$  so that we can represent its polarization state as a Jones vector (see chapter 1, 1.7). In figures 2.7a and 2.7b we see that we



(a) The normal component ( $\sigma$ ) is perpendicular to the plane containing  $\beta_0$  and  $\beta_H$ , so that  $\chi = \pi/2 \implies \sin \chi = 1$ .  
 (b) The parallel component ( $\pi$ ) lies in the plane containing  $\beta_0$  and  $\beta_H$ , so that  $\chi = \pi/2 - 2\theta \implies \sin \chi = \cos 2\theta$ .

Figure 2.7

can set  $\sin \chi = P$ , with

$$\left. \begin{aligned} P &= 1, & \sigma - \text{component} \\ P &= \cos 2\theta, & \pi - \text{component} \end{aligned} \right\}$$

From equation (2.14) and (2.18) and knowing that  $\beta_H^2 = k_0^2(1 + 2\delta_H)$ , we can write equation (2.28) as:

$$\xi_0 \xi_H = \frac{1}{4} k_0^2 P^2 \psi_H \psi_{\bar{H}} \quad (2.30)$$

where

$$2k_0 \xi_i \equiv \beta_i \cdot \beta_i - k_0^2(1 + \psi_0) \quad i = 0, H \quad (2.31)$$

This is the fundamental equation describing the *dispersion surfaces* [20, p. 687].

The surfaces in reciprocal space described by  $\xi_0 = 0$  and  $\xi_H = 0$  are spheres centred, respectively, at the origin  $O$  of the reciprocal lattice and at

point  $H$ . The  $(\xi_0, \xi_H) = (0, 0)$  point  $Q$  is, to terms of the order of the square of  $10^{-5}$ , located at a distance  $k_0 n \approx k_0(1 + \frac{1}{2}\psi_0)$  from both  $O$  and  $H$ .

Following Batterman and Cole [20, p. 687], in proximity of  $Q$  we can approximate these spheres with tangent planes and the dispersion surfaces of equation (2.30) become hyperbolas centred around  $Q$ . The incoming and scattered wavevectors can be determined in a way analogous to the Ewald construction in figure 2.5: for each reciprocal vector  $\mathbf{H}$ , the vectors  $\beta_0$  and  $\beta_H$  can be drawn starting from every point on the hyperbolic surfaces as depicted in figures 2.8 and 2.9.

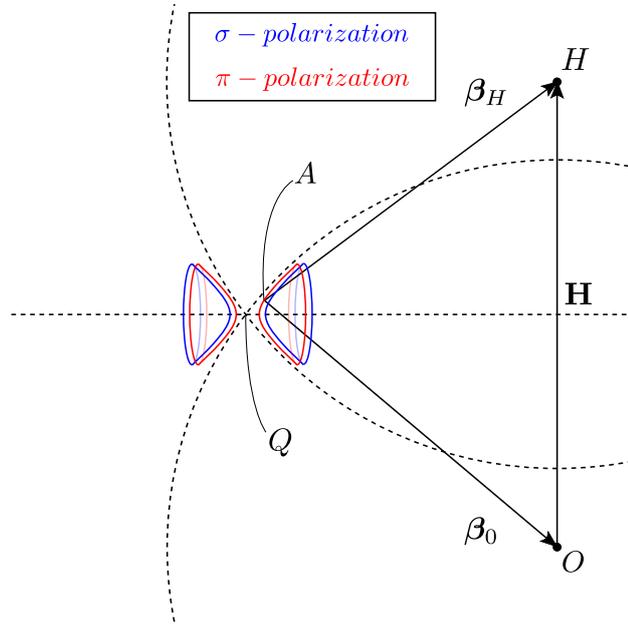


Figure 2.8: In this revised Ewald construction the proportions are exaggerated: the segment  $AH$  should be roughly six orders of magnitude longer than  $AQ$ .

## 2.5 The Bragg & Laue geometries

We have used dispersion surfaces as a graphical tool to determine the direction and modulus of the incident and scattered wavevectors; now we want to go further and find out the amplitude ratio between incident and scattered wave fields<sup>2</sup>.

Going back to equations (2.28) and (2.29) and following the calculations in [10, pp. 117-119], we find that the dispersion equation yields two solutions

<sup>2</sup>The following treatment holds for both the  $\sigma$  and  $\pi$  components, the only difference being the value of the polarization coefficient  $P$ .

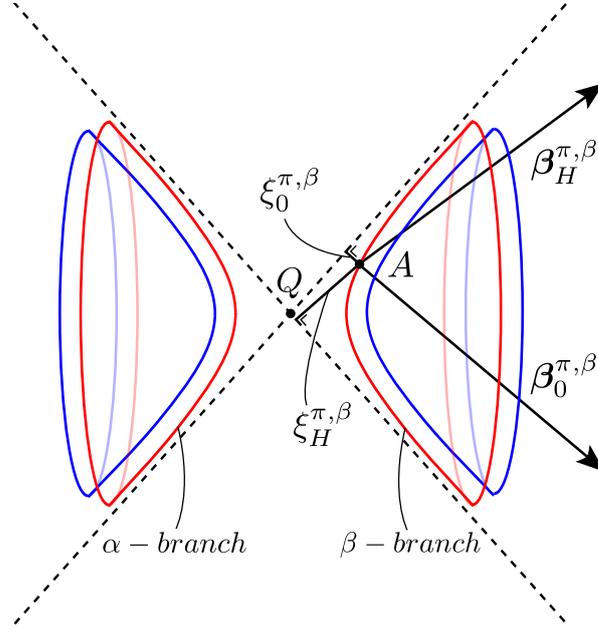


Figure 2.9: The blue and red hyperbolas correspond respectively to the  $\sigma$  and  $\pi$  components of the wave field. A point on one of the dispersion surface branches is called a *tie point*.

for the parameter  $x \equiv D_H/D_0$ :

$$\left. \begin{array}{l} x_1 \\ x_2 \end{array} \right\} = \frac{-z \pm \sqrt{q + z^2}}{\psi_{\bar{H}} P} \quad (2.32)$$

where the parameters  $z$  and  $q$  are defined as:

$$\frac{1}{b} \equiv \frac{\hat{\mathbf{n}} \cdot (\mathbf{k}_0^e + \mathbf{H})}{\hat{\mathbf{n}} \cdot \mathbf{k}_0^e} \approx \frac{\gamma_H}{\gamma_0}, \quad \alpha \equiv \frac{1}{k_0^2} \{H^2 + 2\mathbf{k}_0^e \cdot \mathbf{H}\}, \quad (2.33)$$

$$z \equiv \frac{1-b}{2} \psi_0 + \frac{b}{2} \alpha, \quad q \equiv b \psi_H \psi_{\bar{H}} P^2. \quad (2.34)$$

There are also two possible values for the parameter  $\delta_0$ :

$$\left. \begin{array}{l} \delta_0' \\ \delta_0'' \end{array} \right\} = \frac{1}{2} \{ \psi_0 - z \pm \sqrt{q + z^2} \} \quad (2.35)$$

Thus we can describe the internal fields (both incident and diffracted) in the general case as the sum of the two components which satisfy the self-consistency conditions (2.27):

$$\begin{array}{ll} \text{incident wave field:} & e^{i\mathbf{k}_0^e \cdot \mathbf{x} - i\omega t} (D_0' e^{i\phi_1 d} + D_0'' e^{i\phi_2 d}) \\ \text{diffracted wave field:} & e^{i(\mathbf{k}_0^e + \mathbf{H}) \cdot \mathbf{x} - i\omega t} (x_1 D_0' e^{i\phi_1 d} + x_2 D_0'' e^{i\phi_2 d}) \end{array} \quad (2.36)$$

with

$$\phi_1 \equiv \frac{k_0 \delta'_0}{\gamma_0}, \quad \phi_2 \equiv \frac{k_0 \delta''_0}{\gamma_0}, \quad d \equiv -\hat{\mathbf{n}} \cdot \mathbf{x}. \quad (2.37)$$

Up to now we have considered the interaction between the incident field entering a crystal and the diffracted (or scattered) field it generates within it. A *unique solution* to this problem can be found only by applying adequate *boundary conditions*.

Let us deal with a bounded crystal in the form of a *plane parallel plate* with unlimited lateral extension; the incident wave enters the crystal through the  $\hat{\mathbf{n}} \cdot \mathbf{x} = d = 0$  plane, whilst the diffracted wave can emerge either through the  $d = 0$  or the  $d = d_0$  plane. We distinguish sharply between these two cases which present very different boundary conditions and we treat them separately:

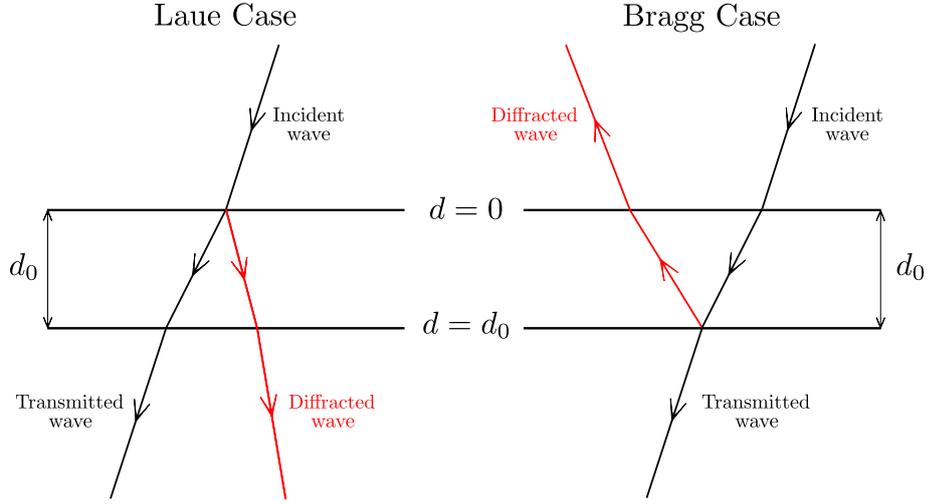


Figure 2.10

### A. The Laue case

The geometry of the Laue case is illustrated in figure 2.10: at the  $d = 0$  surface the *internal* incident field must equal the *external* incident field, whereas the diffracted field amplitude must vanish. These conditions are formalized in the following system:

$$\text{at } d = 0 : \quad \begin{cases} \text{incident field:} & D'_0 + D''_0 = E_0^e \\ \text{diffracted field:} & x_1 D'_0 + x_2 D''_0 = 0 \end{cases} \quad (2.38)$$

This leads to the following solution for the internal fields' intensities:

$$D'_0 = \frac{x_2}{x_2 - x_1} E_0^e, \quad D''_0 = \frac{-x_1}{x_2 - x_1} E_0^e \quad (2.39)$$

which can be inserted in equations (2.36) to yield the intensity ratios at the  $d = d_0$  plane for both the diffracted wave and the *transmitted wave*, i.e. the part of the incident field which propagates out of the crystal without having been diffracted:

$$\frac{\text{diffracted wave field}}{\text{incoming wave field}} = \frac{E_H^d}{E_0^e} = \frac{x_1 x_2 (c_1 - c_2)}{x_2 - x_1} \quad (2.40)$$

$$\frac{\text{transmitted wave field}}{\text{incoming wave field}} = \frac{E_H^t}{E_0^e} = \frac{x_2 c_1 - x_1 c_2}{x_2 - x_1} \quad (2.41)$$

where  $c_1 \equiv e^{-i\phi_1 d_0}$  and  $c_2 \equiv e^{-i\phi_2 d_0}$ . From the amplitude ratio we can calculate both the intensity ratio and the phase.

### B. The Bragg case

We can approach the Bragg case (see figure 2.10) as done with the Laue case, by imposing that the diffracted field amplitude at the  $d = d_0$  plane be zero;

$$\text{diffracted field at } d = d_0 : \quad c_1 x_1 D_0' + c_2 x_2 D_0'' = 0. \quad (2.42)$$

As done in the previous case, we can derive the expressions for the intensity ratios<sup>3</sup>:

$$\text{at } d = 0 : \quad \frac{\text{diffracted wave field}}{\text{incoming wave field}} = \frac{I_H^d}{I_0^e} = \frac{x_1 x_2 (c_2 - c_1)}{x_2 c_2 - x_1 c_1} \quad (2.43)$$

$$\text{at } d = d_0 : \quad \frac{\text{transmitted wave field}}{\text{incoming wave field}} = \frac{I_H^t}{I_0^e} = \frac{c_1 c_2 (x_2 - x_1)}{x_2 c_2 - x_1 c_1} \quad (2.44)$$

Notice that the equations we have found still contain the polarization coefficient  $P$  hidden in the  $x_i$  parameters, so each of the four cases we considered can actually be split in two: one case for the  $\sigma$  and one for the  $\pi$  plane wave component. The general polarization state can then be expressed as a linear combination of the two.

---

<sup>3</sup>In (2.43) we write  $c_2 - c_1$  whereas Zachariassen writes  $c_1 - c_2$ . However, Zachariassen gives the formula for the intensities instead of the amplitudes, so these terms are squared and the results are the same even with a sign difference. In our case the sign cannot be ignored because the phases would be different.

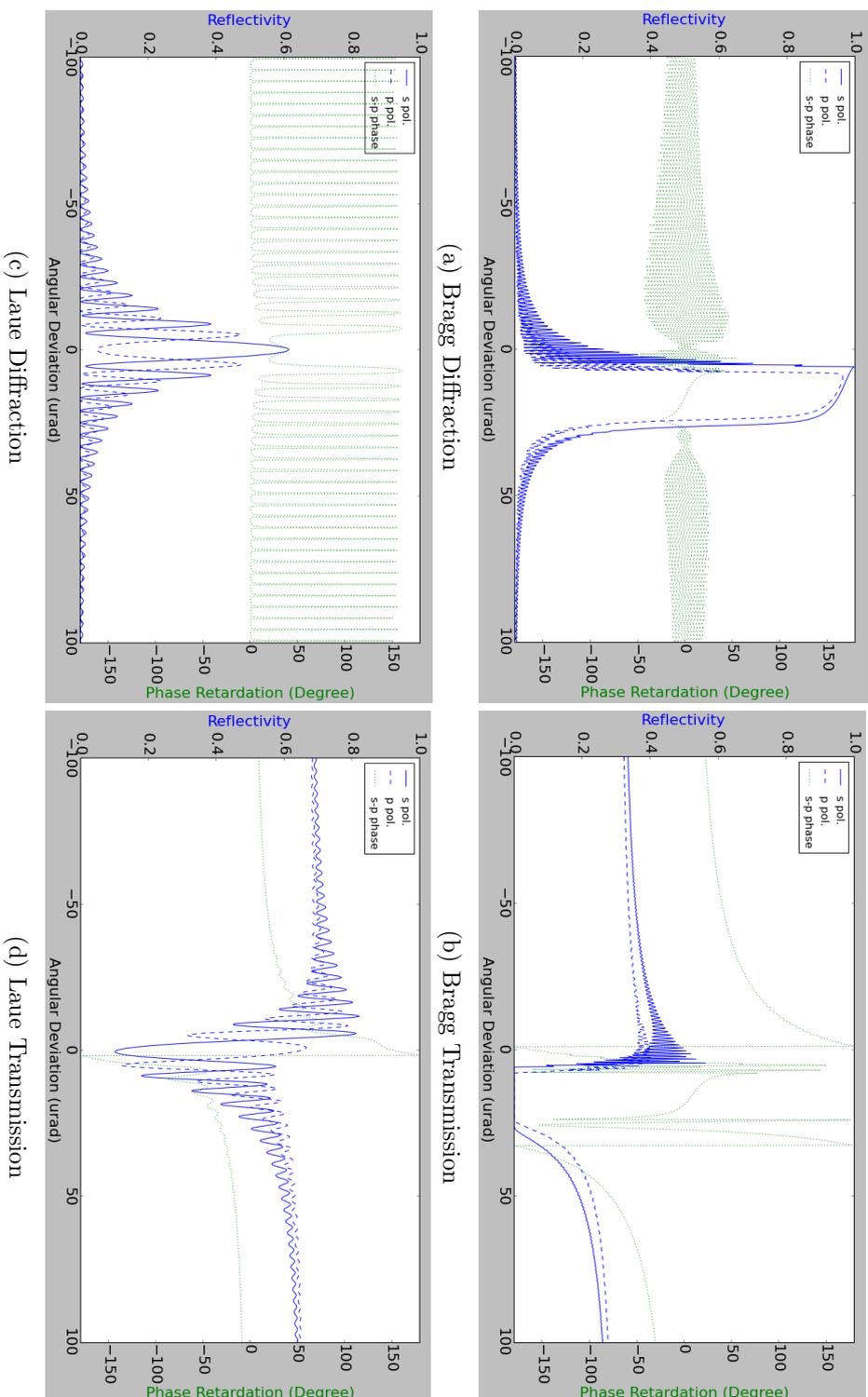


Figure 2.11: Intensities and phase retardation profiles calculated for 10 keV X-rays diffracted by the 220 planes of a 50- $\mu\text{m}$ -thick Si crystal.

## Chapter 3

# The OASYS Graphical Environment for Beamline Optics Design

In 2009 the ESRF has embarked upon an ambitious upgrade programme [21] consisting of two distinct phases:

- *Phase I* (2009-2015) consists of an overhaul of about one third of the beamlines (8 new, 7 fully refurbished) with some being extended to reach source-sample distances of up to 140 meters. A longer beamline allows to reach a higher level of coherence for phase contrast imaging techniques as well as to focus the x-rays into micro and nano beams, achieving a very high demagnification of the ESRF source [22, p. 173].
- *ESRF-EBS* (2015-2022) aims to renovate the whole storage ring in order to reduce electricity costs (20%) and increase both brilliance and coherence by a factor of 100. It will also provide four new state-of-the-art beamlines and new instrumentation able to make the most of the new source's characteristics [23, p. 9].

The new beamlines have unique characteristics, thus a thorough analysis of the performances and an accurate optimization of the design are needed. This, together with similar upgrade programmes under way in other facilities around the world (e.g. the APS Upgrade at Argonne National Laboratory, Illinois) and the construction of entirely new fourth-generation synchrotron light sources (e.g. MAX IV in Lund, Sweden) has prompted a collective effort to improve, renovate and integrate different existing software packages for x-ray optics simulations both inside and outside the ESRF.

### 3.1 Beamline simulation software

When designing a synchrotron beamline, it is essential to resort to computer modelling techniques in order to assess what the criticalities are for the different experiments and address them by optimizing the design. A full beamline simulation routine consists of a study of *source emission*, a thorough *heat load* analysis on the white mirrors and monochromators using finite element methods and *optics* calculations with ray-tracing [24]. Many of the codes used to perform these analyses were developed in academic and research institutions and some of them are open-source.

Let's consider two codes which are well established in the synchrotron community: SHADOW [12] and SRW [25]. They both have the goal of simulating x-ray propagation along the beamline, but they do it through different models: SHADOW uses geometrical optics (ray tracing) while SRW uses physical optics (wavefront propagation from single electrons). Ray tracing provides incomplete functionality when diffraction effects due to finite apertures become relevant, while wavefront propagation methods based on scalar diffraction theory become computationally cumbersome when dealing with partially coherent sources modelled as emitted from incoherent "micro-electrons" (which is the case for synchrotron radiation)[26]. A user might then want to harness the advantages of both methods by using both to get complementary information and to do it in a compatible mode, with the same user interface and allowing for information exchange between the two.

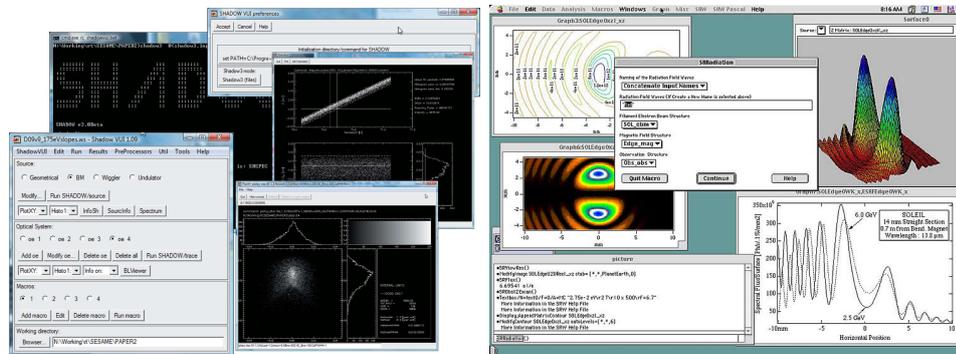


Figure 3.1: Both SHADOW (left) and SRW (right) have their own GUIs.

Apart from these two, a host of other codes are available: *McXtrace* for Monte Carlo ray tracing [27] and *XRayTracer* (xrt) [28] are among the best known in the community. Additional algorithms have also been proposed to be integrated with SHADOW or other ray tracing tools with the aim of getting around some of their shortcomings: notable examples are the *hybrid method* [26] and *coherent-mode decomposition*, which is currently being implemented by Mark Glass at ESRF.

In order to make sense of this jungle of different software, packages and algorithms, enhancing collaborations between the developers and making it possible for users to operate several codes side-by-side in an interactive fashion, with a common interface and notation, OASYS has been created.

## 3.2 Virtual experiments in OASYS

OASYS (OrAnge SYnchrotron Suite) is an Open Source Graphical Environment developed by Manuel Sanchez del Rio (ESRF) and Luca Rebuffi (ELETTRA) [29]. Its goal is to model a *virtual experiment* by integrating different programs into a single user-friendly GUI (Graphical User Interface) to give long term support to synchrotron facilities in beamline simulations.

A *virtual experiment* consists in modeling and simulating a whole synchrotron experiment from storage ring dynamics to light-matter interaction in the detectors. It requires specific codes able to tackle all the different steps involved, namely:

- calculation of the electron beam parameters based on a detailed description of the storage ring lattice (electromagnetic lenses, bending magnets, insertion devices, RF cavities et cetera);
- calculation of the radiation emitted by the electrons when accelerated by magnetic fields in bending magnets, undulators or wigglers, based on the electron beam parameters from the previous step;
- simulation of how beamline optical elements (mirrors, gratings, crystals, multilayers, lenses et cetera) determine the propagation and properties of the emitted x-ray radiation;
- simulation of the interaction with samples and detectors.

We can then see that the virtual experiment is made up of a sequence of calculations where each step (undulator, mirror, lens, sample . . .) determines the input to be fed to the next. The system can then be modeled as a *workflow* composed of a chain of components each receiving and/or sending out a beam or "signal" which contains all the data required for the next step, these may be information about electron beam parameters or photon energy, polarization, coherence of the photon beam, divergences et cetera.

Python was chosen as interaction language because of its popularity in the scientific community which stems from (and has led to) a high number of tools and libraries being developed in Python so that many programs (SHADOW and SRW among others) have a Python Application Program Interface.

Orange was developed at the University of Ljubljana as an open source platform for data mining [30] and was chosen as GUI because of its adapt-

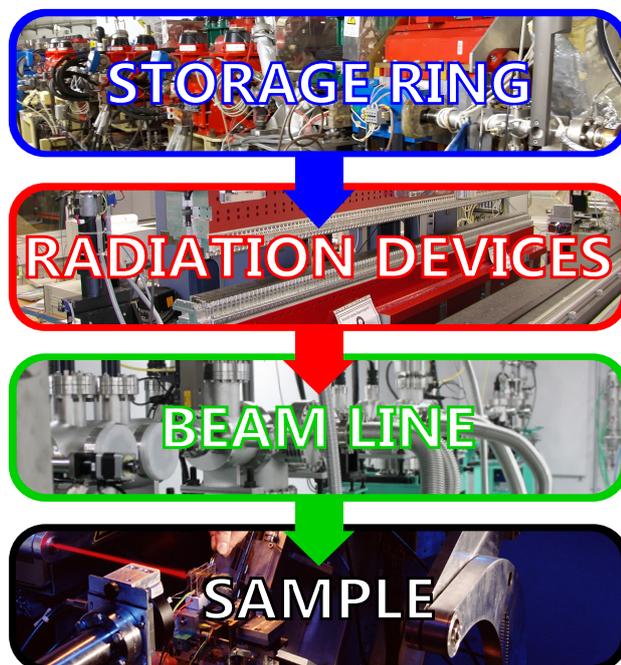


Figure 3.2: The main steps of the virtual experiment

ability and its beautiful workflow based on connected widgets which truly resembles an actual optical system.

At the moment (March, 2017) a beta release for OASYS is already available which includes *ShadowOui*, the OASYS user interface for SHADOW3. More and more features are going to be added in the coming months and years and several workshops have already been organised (e.g. [31]) to allow both users and programmers to familiarize with this new platform in the hope of creating a broad and bustling community.

The work in this thesis is driven by the need of some particular beam-lines to perform calculations with polarized beams and to study the effect of crystal phase shifters. A code called *polarsim* was written by Christian Vettier in the 1980's and has been available at the ESRF for many years. It was used in early works of crystal phase plates [32, 33]. The functionalities of *polarsim*, which also included Müller-Stokes analysis, are now implemented and improved in the new *crystalpy* package in OASYS.

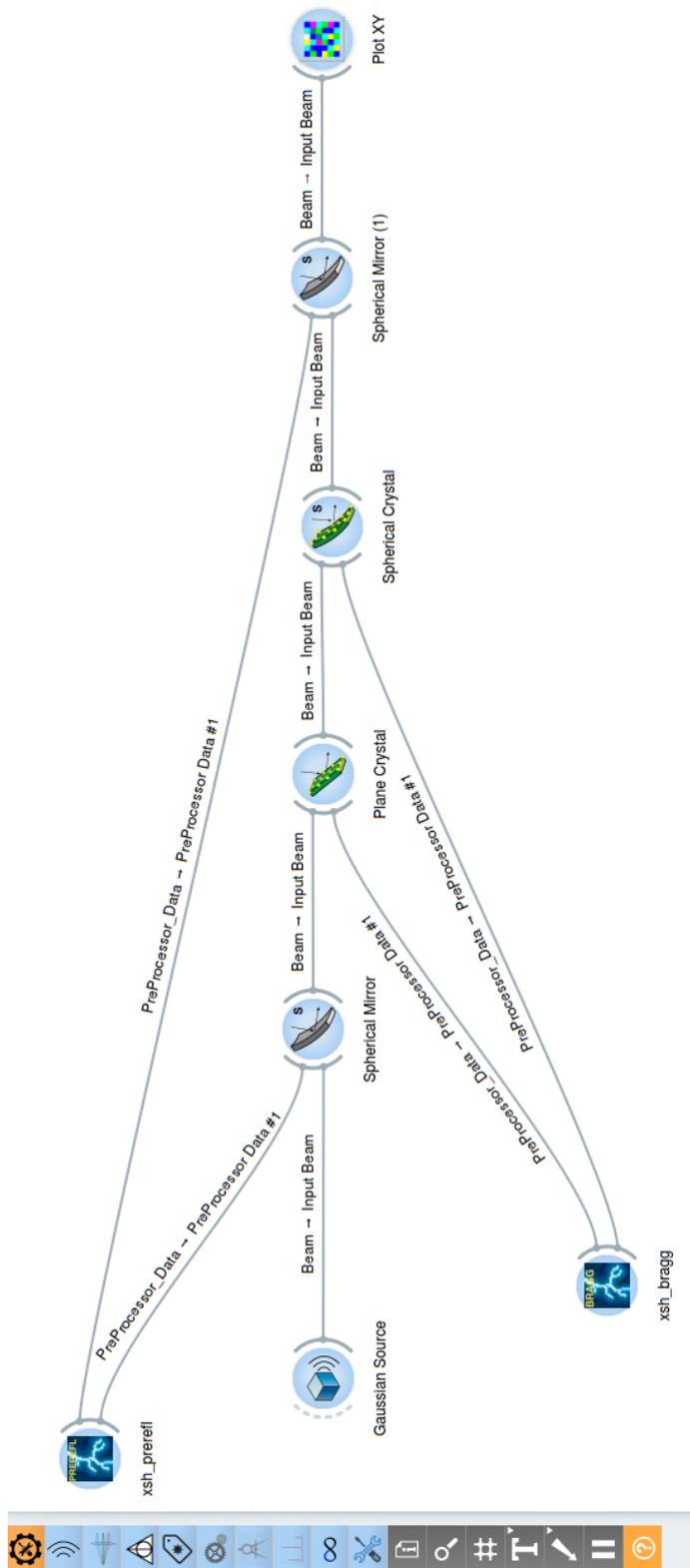


Figure 3.3: Example of an OASYS workflow on the Orange canvas showing ShadowOui widgets.

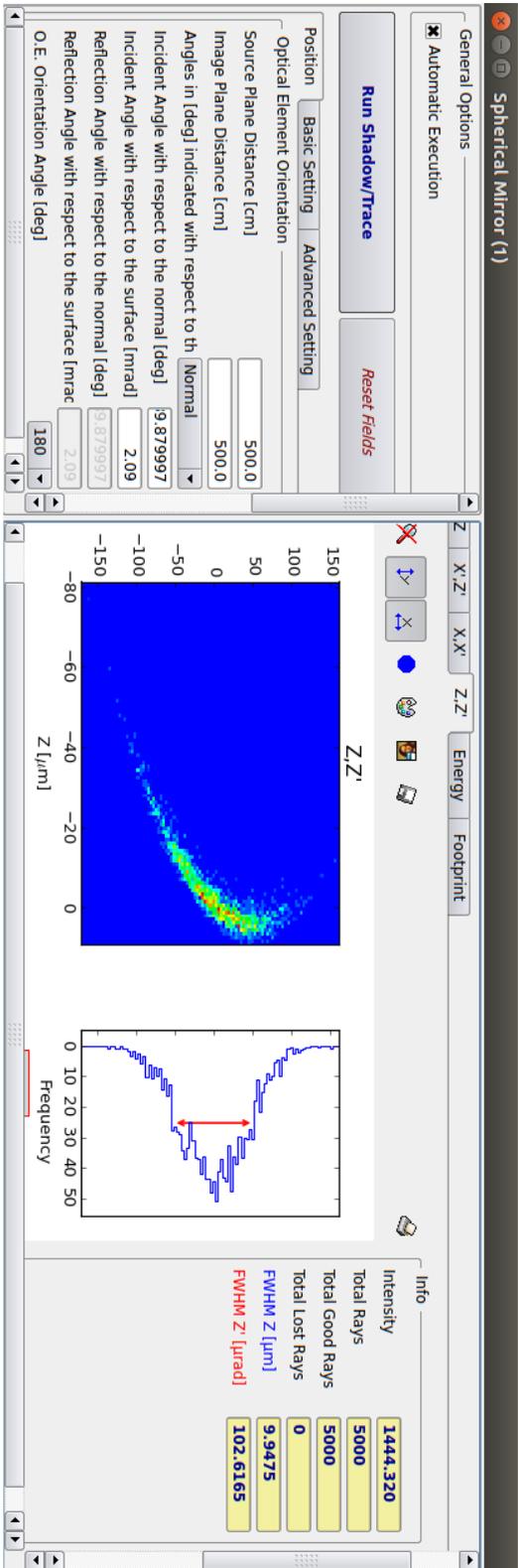


Figure 3.4: Example of a ShadowOui spherical mirror widget: on the left the user can set the mirror parameters, while on the right it is possible to see the beam's spot size and intensity profile downstream of the mirror.

## Chapter 4

# The crystalpy Library

`crystalpy` is a Python library which performs calculations on diffraction from perfect crystals using the Zachariasen formalism (see chapter 2) and simulates the changes in the polarization state of the x-ray beam brought about by the diffraction using Müller-Stokes formalism. The library was developed mainly by myself and Mark Glass with valuable insight, suggestions and contributions by Manuel Sanchez del Rio. It is divided into three directories: `util`, `diffraction` and `polarization`<sup>1</sup>.

### 4.1 The util toolbox

The `util` toolbox contains auxiliary classes which take part in the diffraction process but have their own significance independently of it.

- The `Vector` class represents a physical vector containing methods for addition, scalar product, cross product, normalization, rotation around an axis et cetera. This object is fundamental as the whole library was written with the idea of making everything work with vectorial quantities instead of using angles. To clarify the advantages of this approach, let us imagine we have an x-ray beam impinging on the surface of a crystal at an angle  $\theta$  with respect to the normal: how do we describe this in the code? The simplest way would be to identify its direction by assigning  $\theta$  to a variable, e.g. `angle_with_normal`, while a more complicated way would be to use an object-oriented approach and write:

```
normal = Vector(0, 0, 1) # z axis
beam_direction = normal.rotateAroundAxis(Vector(1, 0, 0), theta)
```

This is more cumbersome, both at design and run time, but it allows for a very straightforward manipulation of the vectors: if, for example,

---

<sup>1</sup>The code can be found at <https://github.com/edocappelli/crystalpy>.

we wanted to know the angle between the beam direction and the reciprocal vector  $H$ , we would also have to keep track of the angle between  $H$  and the normal and pay attention to the sign of the angles in order to avoid mistakes. The object-oriented approach becomes even more advantageous in a three-dimensional system, where storing one angle is not enough and finding the angle between vectors that do not share the same plane with the normal can be quite a head-scratcher<sup>2</sup>.

- The `StokesVector` class has the four Stokes parameters as class attributes and contains a method for calculating the degree of circular polarization, defined as  $P_c = S_3/S_0$ .
- The `Photon` class is characterised by an energy (in eV) and a direction vector. The `PolarizedPhoton` class inherits from it and has a `StokesVector` attribute in addition to the two attributes of the parent class. It is an abstraction of a monochromatic plane wave rather than of a physical photon.
- Both `Photon` and `PolarizedPhoton` have a corresponding `PhotonBunch` object, which is simply an *iterable* containing a collection of photons<sup>3</sup>, this means it can be indexed and called iteratively by using the `in` keyword:

```
for photon in PhotonBunch:
    do_something()
```

These classes are used as "beam" or "signal" between different widgets when integrating the library into OASYS.

## 4.2 The diffraction toolbox

This part of the library is concerned with the calculation of x-ray diffraction from perfect crystals as discussed in chapter 2 using the formulation from Zachariasen [10]. Let us explore this toolbox and see how the computation is handled.

1. The `DiffractionSetup` class handles the information about the crystal setup and collects all the parameters needed to fully define the physical system we are modelling:

```
geometry_type    clarifies which is the diffraction geometry among
                  BraggDiffraction, BraggTransmission, LaueDiffraction and
                  LaueTransmission. These are the four cases we distinguished in
                  chapter 2, section 2.5.
```

---

<sup>2</sup>The `rotateAroundAxis` function is implemented using *Rodrigues' formula* [34].

<sup>3</sup>Please notice that the term *photon* will be used here to refer to a `Photon` object even if it does not correspond to a physical photon.

`crystal_name` is a string (e.g. Si, Ge ...) specifying which crystal is being used for the diffraction.

`thickness` of the crystal in centimetres; we are considering parallel plane plates, so this parameter is always well defined.

`millers_h(k,l)` are the Miller indices identifying the family of lattice planes which satisfy the Laue condition. Remember that in the two-wave approximation only one family of planes takes part in the diffraction process, so three indices are enough to fix the  $H$  vector once the crystal structure is known.

`asymmetry_angle` is shown graphically in figure 4.1 and is the angle between the crystal surface and the planes  $hkl$  as defined in [35]. It depends on how the crystal has been cut.

It takes advantage of the functions implemented in the `xraylib` library [36] to provide methods able to yield important parameters such as the *Bragg angle* at a certain energy: `angleBragg(energy)`, the *structure factor* Fourier components  $F_0$ ,  $F_H$  and  $F_{\bar{H}}$ : `F0(energy)`, `FH(energy)`, `FH_bar(energy)`, the *unit cell volume*: `unitCellVolume()`, the lattice spacing for the  $hkl$  family of planes: `dSpacing()` and the reciprocal vector  $\mathbf{H} = h\mathbf{b}_1 + k\mathbf{b}_2 + l\mathbf{b}_3$ : `normalBragg()`. If one just wants to plot the reflectivities/transmittivities as functions of angular deviation or energy, the `DiffractionSetupSweeps` class can be used to build a `PhotonBunch` containing all the photons for whom the calculations are going to be carried out. It needs to know how many photons to create and in which angular and/or energy range, so it takes as attributes `energy_min`, `energy_max`, `energy_points` for the energy and `angle_deviation_min`, `angle_deviation_max` and `angle_deviation_points` for the angular deviations from the  $(0, 1, 0)$  direction.

2. The `Diffraction` class manages the computations without actually doing anything at initialization. It behaves as an administrator, directing the inputs to where they are needed and collecting the results in an orderly fashion to create the output. The advantage of this solution is that, in order to compute the results from the main, an Orange widget or a user script, one only needs to do this:

```
diffraction = Diffraction()
outgoing_photon_bunch = diffraction.calculateDiffractedPhotonBunch(
    diffraction_setup,
    incoming_photon_bunch)
```

The `calculateDiffractedPhotonBunch` method iterates over all the photons in the incoming bunch and for each one calls the `calculateDiffractedPhoton` function. This in turn creates an instance of the

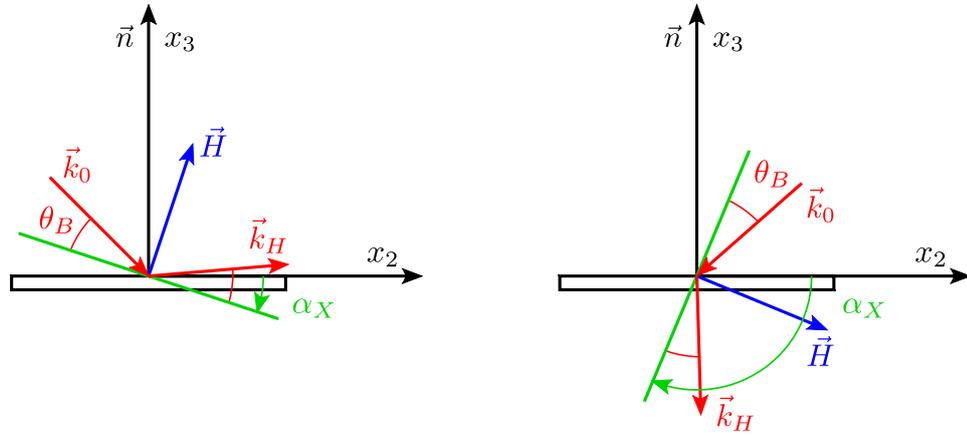


Figure 4.1: Schematic drawings of Bragg (left) and Laue (right) diffraction setups:  $\mathbf{H}$  is the reciprocal vector associated with the  $hkl$  family of lattice planes, also called the *Bragg normal*,  $\theta_B$  is the *Bragg angle* and  $\alpha_X$  is the *asymmetry angle* as defined in [35].

`PerfectCrystalDiffraction` class. This class contains methods to perform the actual calculations. This matryoshka-like structure is illustrated in figure 4.2. `Diffraction` is designed to propose different methods of calculation, or different algorithms. At present, only Zachariasen theory for perfect crystals is available.

3. `PerfectCrystalDiffraction` is the class which handles the actual computations. It calculates the Zachariasen parameters by means of several methods and then plugs them into the formulae 2.40, 2.41, 2.43 or 2.44 according to the specified experimental geometry: for each one setup the results are two amplitude ratios for the two polarization states S (i.e.  $\sigma$ ) and P (i.e.  $\pi$ ). A graphical representation of the steps of the calculation is shown in figure 4.3.

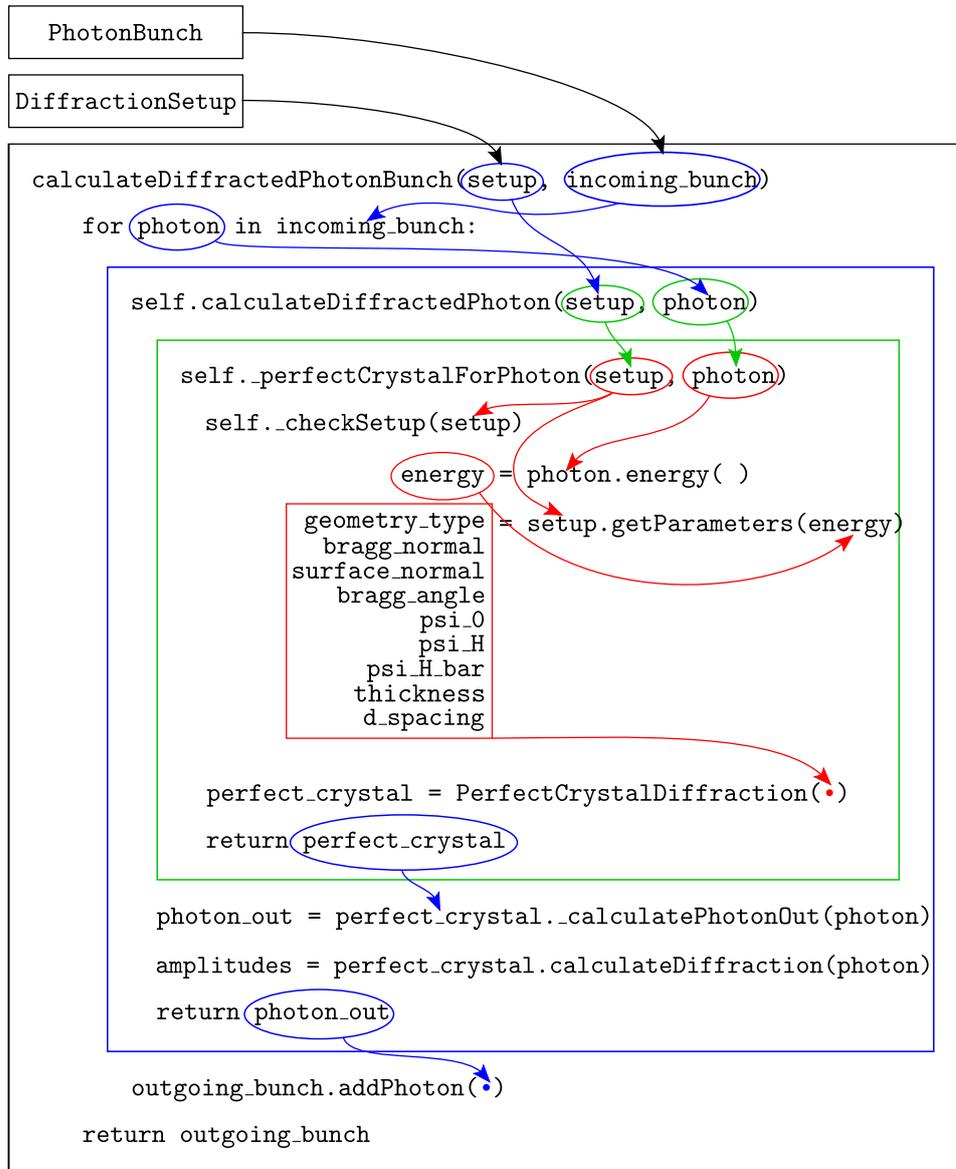


Figure 4.2: Structure and data flow of the `Diffraction` class. The `_checkSetup` method checks whether the parameters from `DiffractionSetup` are not contradictory, e.g. it verifies that, if the declared geometry type is Bragg, the asymmetry angle be smaller than the Bragg angle (why it must be so becomes clear by inspecting figure 4.1); if this is not true, then the angles are not compatible with the expected geometry and an exception must be raised. We will see when dealing with the `polarization` toolbox that the amplitudes are used to generate a "polarized photon" as output. The  $\psi$  parameters are computed via equation 2.21.

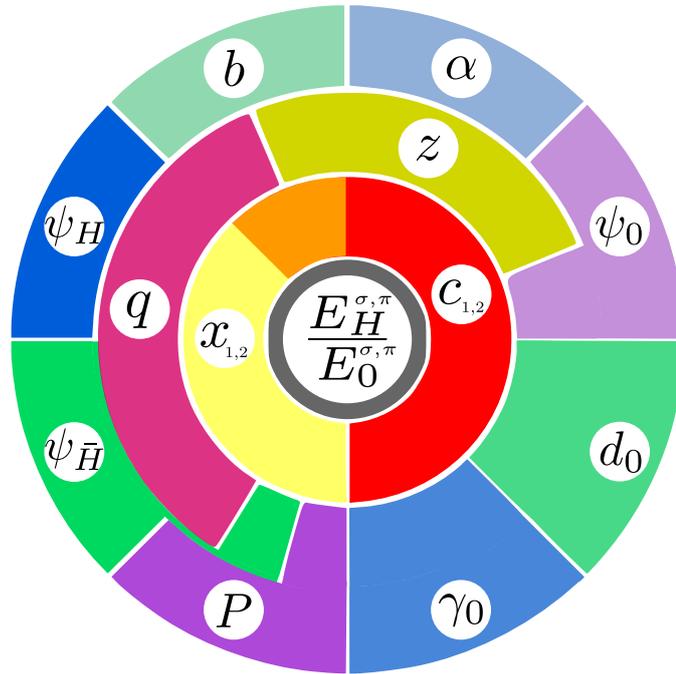


Figure 4.3: This is a board-game-style representation of the calculation process in the `PerfectCrystalDiffraction` class: the letters stand for the parameters used by Zachariassen which are computed using the equations we have derived in chapter 2. Each one of the concentric circles is one "layer" of computation, i.e. the innermost parameters are computed starting from the parameters surrounding them on the outside. For example, to obtain  $z$  using equation 2.34 we need  $b$ ,  $\psi_0$  and  $\alpha$ , which are adjacent to  $z$  and located further out; to compute  $x_{1,2}$  via 2.32 we need  $z$ ,  $q$ ,  $\psi_{\bar{H}}$  and  $P$ , and so on.

### 4.3 The polarization toolbox

This toolbox is made up of classes needed to allow `Diffraction` to perform polarization calculations through class methods such as `calculateDiffractedPolarizedPhotonBunch` and `calculateDiffractedPolarizedPhoton`, which follow the same process described in figure 4.2 but have `PolarizedPhoton` objects as inputs and outputs. These functions compute `photon_out` and `amplitudes` through the dedicated `PerfectCrystalDiffraction` methods and then use the field amplitudes to build the Müller matrix for a crystal phase plate. The matrix is then applied to the Stokes vector attribute of the incoming `PolarizedPhoton` object to get the outgoing polarized photon. The single Stokes parameters can then be extracted and plotted as a function of angular deviation from the Bragg angle to analyse the birefringence of the crystal phase plates.

To do this the toolbox contains:

- The `MuellerMatrix` class has a `matrix` attribute which contains a  $4 \times 4$  numpy matrix. It has a few methods which allow it to behave as a Müller matrix should: `matrix_by_scalar(scalar)` for scalar multiplication, `mueller_times_mueller(matrix)` for matrix multiplication, and `calculate_stokes_vector(incoming_stokes_vector)` to apply the matrix to an incoming Stokes vector to get an outgoing Stokes vector.
- The `CrystalPhasePlate` class inherits from `MuellerMatrix`, since the matrix for a specific phase plate is nothing but a particular instance of the abstract concept of a Müller matrix.

The crystal plate's reference frame will in general be rotated with respect to the source frame as shown in figures 4.4 and 4.5. The Jones matrix for the phase plate is then made up of a rotation by an angle  $\alpha$ , called the *inclination angle*, and the subsequent crystal diffraction:

$$\begin{aligned} \begin{pmatrix} E_\sigma^t \\ E_\pi^t \end{pmatrix} &= \begin{pmatrix} t_\sigma e^{i\phi_\sigma} & 0 \\ 0 & t_\pi e^{i\phi_\pi} \end{pmatrix} \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} E_\sigma \\ E_\pi \end{pmatrix} \\ &= \begin{pmatrix} t_\sigma e^{i\phi_\sigma} \cos \alpha & t_\sigma e^{i\phi_\sigma} \sin \alpha \\ -t_\pi e^{i\phi_\pi} \sin \alpha & t_\pi e^{i\phi_\pi} \cos \alpha \end{pmatrix} \begin{pmatrix} E_\sigma \\ E_\pi \end{pmatrix} \end{aligned} \quad (4.1)$$

This result can then be translated into the more general Müller-Stokes formalism and we can do so by applying equation 1.21, which yields:

$$\begin{aligned} \mathbf{s}' &= \mathcal{M}_{BT} \cdot \mathbf{s} \\ &= \begin{pmatrix} (t_\sigma^2 + t_\pi^2)/2 & (t_\sigma^2 - t_\pi^2) \cos 2\alpha/2 & (t_\sigma^2 - t_\pi^2) \sin 2\alpha/2 & 0 \\ (t_\sigma^2 - t_\pi^2)/2 & (t_\sigma^2 + t_\pi^2) \cos 2\alpha/2 & (t_\sigma^2 + t_\pi^2) \sin 2\alpha/2 & 0 \\ 0 & -t_\sigma t_\pi \cos \Phi \sin 2\alpha & t_\sigma t_\pi \cos \Phi \cos 2\alpha & -t_\sigma t_\pi \sin \Phi \\ 0 & -t_\sigma t_\pi \sin \Phi \sin 2\alpha & t_\sigma t_\pi \sin \Phi \cos 2\alpha & t_\sigma t_\pi \cos \Phi \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} \end{aligned} \quad (4.2)$$

where  $\Phi = \phi_\pi - \phi_\sigma$  is the *phase difference* between the two field components.

At initialization time the class needs five parameters: `intensity_sigma`, `intensity_pi`, `phase_sigma`, `phase_pi` and `inclination_angle`. With these it computes all the entries of the matrix at 4.2, writes them on a numpy matrix and then calls the constructor of the parent class with this matrix as parameter.

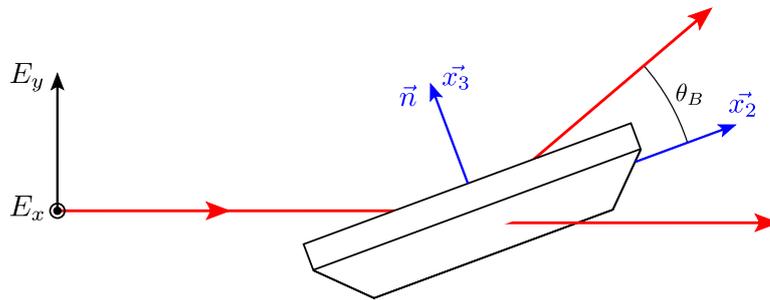


Figure 4.4

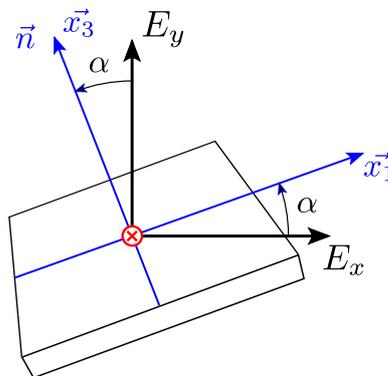


Figure 4.5

The `Diffraction.calculateDiffractedPolarizedPhoton` method uses these classes as follows:

1. It reads intensities and phases for both  $\pi$  and  $\sigma$  polarizations from the `amplitudes` dictionary;

```
intensity_pi = amplitudes["P"].intensity()
intensity_sigma = amplitudes["S"].intensity()
phase_pi = amplitudes["P"].phase()
phase_sigma = amplitudes["S"].phase()
```

2. It then uses them to create a `CrystalPhasePlate` object;

```
phase_plate = CrystalPhasePlate(
    intensity_sigma=intensity_sigma,
    phase_sigma=phase_sigma,
    intensity_pi=intensity_pi,
    phase_pi=phase_pi,
    inclination_angle=inclination_angle)
```

3. It applies the phase plate Müller matrix to the incoming wave's Stokes vector and calculates the outgoing vector;

```

outgoing_stokes_vector =
    phase_plate.calculate_stokes_vector(incoming_stokes_vector)

```

4. It returns a `PolarizedPhoton` object which has the energy and direction of `outgoing_photon` and the computed Stokes vector;

```

outgoing_polarized_photon = PolarizedPhoton(
    energy_in_ev=outgoing_photon.energy(),
    direction_vector=outgoing_photon.unitDirectionVector(),
    stokes_vector=outgoing_stokes_vector)

```

## 4.4 The oasys-crystalpy graphical interface

The `oasys-crystalpy` library aims to integrate the `crystalpy` functionalities into the OASYS Graphical Environment through several Orange widgets. The ultimate goal is to provide a tool that can easily be used together with SHADOW and become a part of a full-beamline simulation<sup>4</sup>.

`CrystalCalculator` and `CrystalViewer` were developed as stand-alone widgets, giving the user a fast way to get crystal diffraction results and plot them. The first takes from the users all the parameters needed to define the diffraction setup (crystal parameters, diffraction geometry et cetera) as well as the incoming photons in a grid-like source (energy range, angular deviation range, number of points, Stokes vector) while the second plots the results.

The idea behind the remaining widgets was to separate the photon-creation part from the crystal diffraction part as it is the case in a real experiment and as is required for integration with other OASYS tools. This resulted in a more complex but far more versatile structure in which the different widgets perform more specialized tasks and exchange information in the form of `PhotonBunch` objects. The photon bunch is defined in a coordinate system we are going to refer to as the *lab frame*, whereas the crystal parameters are defined in the *crystal frame*, in which  $x_1$  and  $x_2$  are on the crystal surface plane and  $x_3$  is parallel to the surface normal, as shown in figures 4.4 and 4.5.

---

<sup>4</sup>The full code can be found at <https://github.com/edocappelli/oasys-crystalpy>.

## 4.5 The oasis-crystalpy widgets



**PhotonSource** generates a polarized photon bunch with grid-like energy and deviation distributions around the  $x_2 = (0, 1, 0)$  direction in the lab frame, letting the user specify the ranges, number of points and the Stokes parameters' values. A Stokes vector  $(1, 1, 0, 0)$  corresponding to horizontal linear polarization indicates that the electric field is oscillating along the  $x_1$  direction, while the vertical direction is  $x_3$ .



**BendingMagnet** simulates the Stokes parameters for bending magnet radiation starting from the electron beam and magnet parameters using the results by Sokolov and Ternov [37] described in Appendix A.

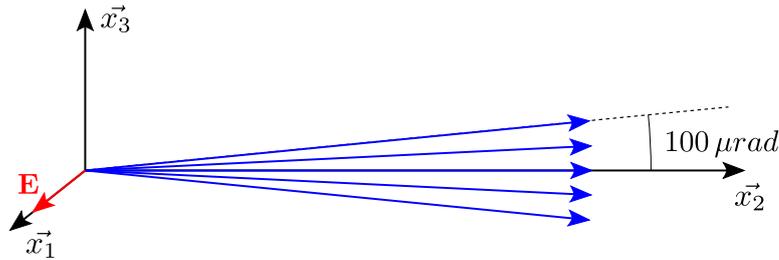


Figure 4.6: Generated photon bunch with default settings: maximum deviation  $100 \mu rad$ , energy 8000 eV, Stokes vector  $(1, 1, 0, 0)$ .



**AlignmentTool** rotates the photon bunch so that a photon with an `energy=base_energy` will impinge on the selected family of crystal planes at an angle  $\theta_{Bragg}$  (see figure 4.1). To do this it needs to know the Miller indices of the diffraction planes, the base energy and the asymmetry angle  $\alpha$ . It acts as an interface between the lab frame and the crystal frame, setting the crystal up according to the specified parameters. It can work both ways and can be used in two modes: `lab-to-crystal` and `crystal-to-lab`.



**Crystal** allows the user to set the diffraction parameters, takes an incoming polarized photon bunch as input and sends out another polarized photon bunch. It uses the `crystalpy` library to generate the diffracted bunch according to the user's specifications.



**ShadowConverter** was written by Manuel Sanchez del Rio to convert a `PolarizedPhotonBunch` object into a Shadow Beam and vice versa, making it possible to use the `crystalpy` library in a beamline simulation along with SHADOW tools.



**PhotonViewer** takes a `PolarizedPhotonBunch` as input and plots the Stokes vectors as functions of angular deviation or energy using `matplotlib` [38] plotting tools.



Figure 4.7: Example of an OASYS workflow for crystal phase plate simulations.

## 4.6 Anatomy of an Orange widget

The goal of OASYS is to create an open source project to which users can add tools or scripts with relative ease, therefore some effort was put into creating some basic widget templates which can be edited even with little understanding of the underlying Orange structure. The `crystalpy` widgets were created with the following structure:

1. The widget must be defined as a child class of the `widget.OWidget` class. We will use the `AlignmentTool` widget as an example.

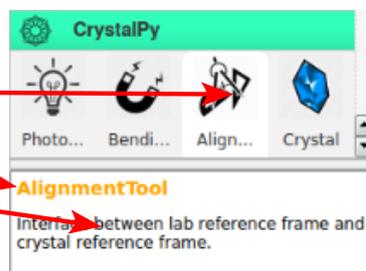
```
class OAlignmentTool(widget.OWidget):
```

2. Some basic information is specified, e.g. widget name, widget description, path to the icon file, author, maintainer e-mail et cetera.

```
icon = "icons/Alignment.png"
```

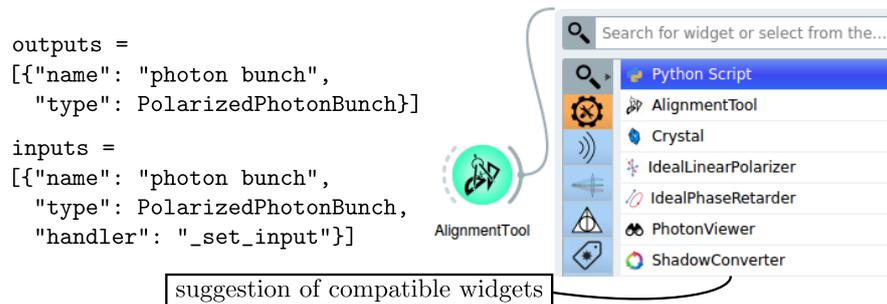
```
name = "AlignmentTool"
```

```
description = "Interface between..."
```

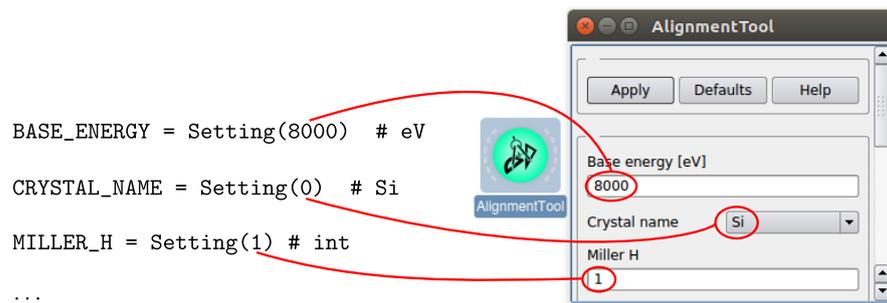


3. Inputs and outputs are defined: one has to include name and object type for both. An input also requires a *handler* method which

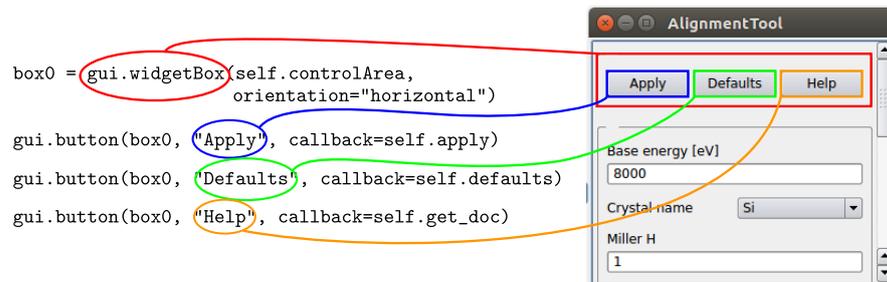
should assign it to a class attribute and, for example, check whether the input is within a certain range of values or do some initial processing. The specification of input and output object types allows Orange to know which widgets are compatible with one another and to suggest which ones the user might want to link together. The program does not allow the user to link widgets if output and input types do not match, thereby avoiding potential runtime errors.



- The default values are set by creating an instance of the Orange `Setting` class. For combo-boxes (the choice of crystal type in this example) one has to specify an array of strings from which to choose from and the default is specified via the array index.

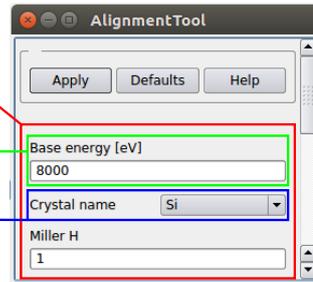


- The "Apply", "Defaults" and "Help" buttons are created in a dedicated box. Each one calls a specific class method: an `apply` method to align the photon bunch and generate the output, a `defaults` method to reset the parameter to the default values, or a `get_doc` method which shows the user a help text file.



- The user can insert the setup parameters using different kinds of interfaces e.g. `lineEdit`, where the user can type the value, or `comboBox`, where the user can choose one out of the available options. With a `lineEdit` interface one can make sure the user can only type in the right kind of value by specifying the `valueType` parameter.

```
box = gui.widgetBox(self.controlArea,
                   orientation="vertical")
gui.lineEdit(box, "BASE_ENERGY",
             valueType=int)
gui.comboBox(box, "CRYSTAL_NAME",
            items=['Si', 'Ge', 'Diamond'])
```



- When the "Apply" button is pushed, the `apply` method creates an outgoing photon bunch object and sends it out through the `send` method. The output object is now available to the widgets downstream.

```
self.send("photon bunch", outgoing_bunch)
```

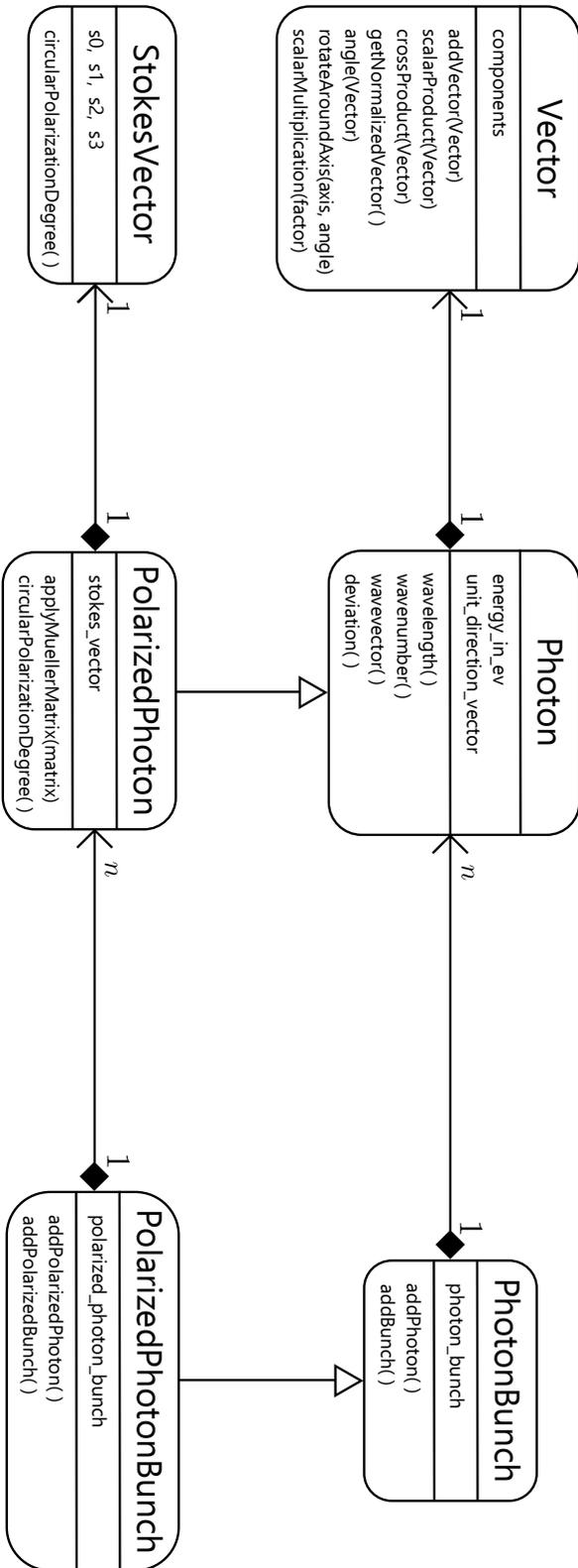


Figure 4.8: UML class diagram for the util toolbox.

# Chapter 5

## Results

In order to assess the program's performance, some tests were conducted against SHADOW and computations/measurements found in existing literature. In the following we will look at some example of these comparative tests.

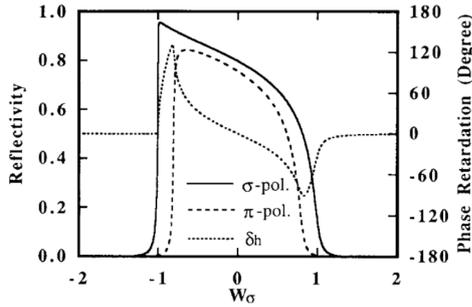
### 5.1 Testing against Hirano *et al.*

A 1993 article by Hirano *et al.* on "Perfect crystal X-ray phase retarders" [4] offers an in-depth analysis of the use of x-ray phase plates to produce circularly polarized synchrotron radiation based on several calculated intensity and phase profiles for different experimental geometries and crystal types. Since the early stages of the development the code has been tested using some of these results as controls, leading to the detection of several bugs and to substantial improvements.

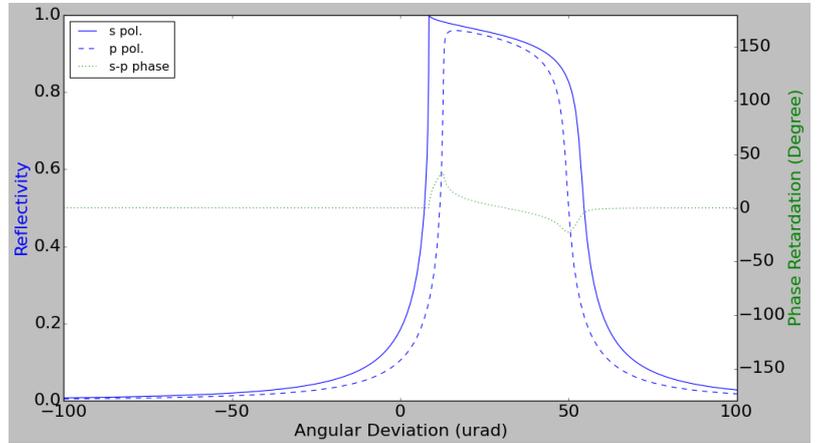
The figures on the left are those by Hirano *et al.*, those on the right are the corresponding `crystalpy` simulations. Notice how Hirano uses the parameter  $W_\sigma$  on the horizontal axis, whereas `crystalpy` uses the angular deviation from the Bragg condition measured in  $\mu rad$ . Since  $W_\sigma$  relates linearly to the deviation  $\delta\theta = \theta - \theta_{Bragg}$  [4, p. 344], we should expect Hirano's diffraction peaks' shapes to be contracted and shifted with respect to the `crystalpy` ones. Indeed, Hirano's figures are all centered at  $W_\sigma = 0$ , while the diffraction peak is actually displaced from the Bragg direction as is visible in figure 5.1.

The oscillations in the phase difference in figures 5.2 and 5.3 are caused by the fact that the function is wrapped within the interval  $[-\pi, \pi]$  while diverging to infinity. Since there is a divergence, by adding more sample points to the plot we get faster and faster oscillations as shown in figure 5.4.

## Ge 220 Bragg diffraction at 10 keV



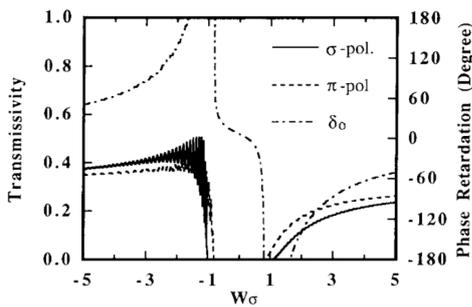
(a) Hirano fig. 3



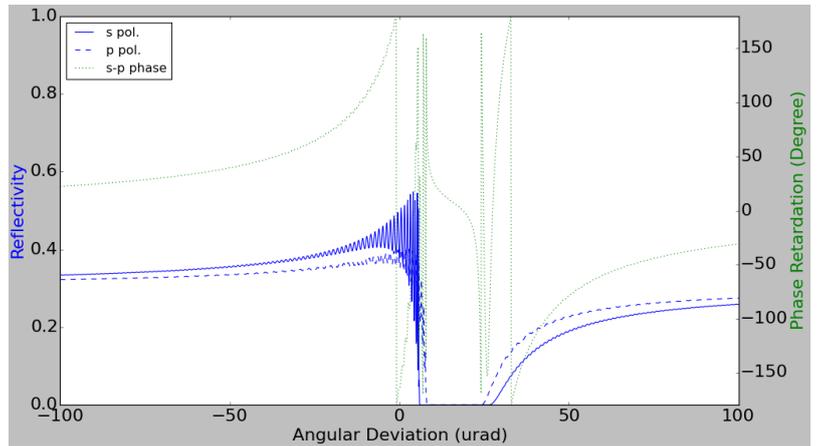
(b) CrystalViewer plot

Figure 5.1: Calculated reflectivities for  $\sigma$  (s) and  $\pi$  (p) polarizations and phase retardation (s - p) for Ge 220 Bragg diffraction at 10 keV. Figure (a) is actually the result of 4 successive Ge 220 reflections, so the cumulative phase difference is bigger in (a) than in (b).

## Si 220 Bragg transmission at 10 keV



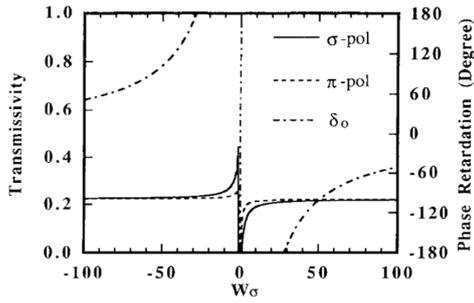
(a) Hirano fig. 4



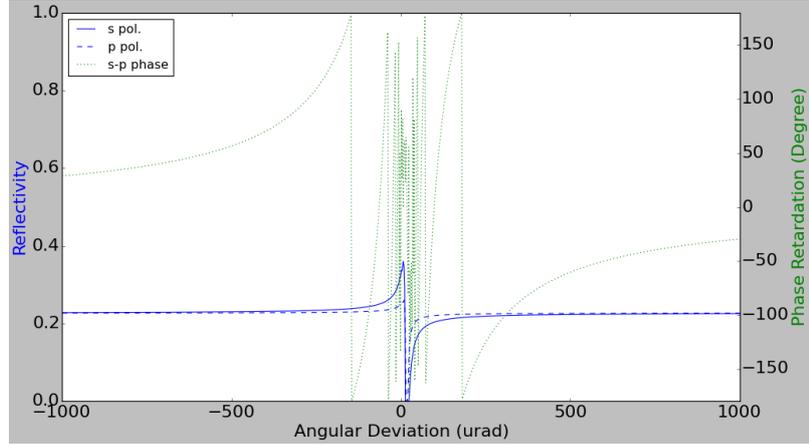
(b) CrystalViewer plot

Figure 5.2: Calculated transmittivities for  $\sigma$  (s) and  $\pi$  (p) polarizations and phase retardation (s - p) for Si 220 Bragg transmission at 10 keV from a crystal  $50\text{-}\mu\text{m}$  thick.

## Diamond 220 Bragg transmission at 10 keV

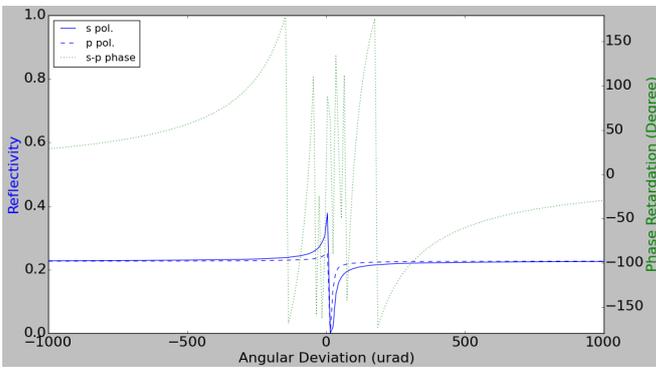


(a) Hirano fig. 5

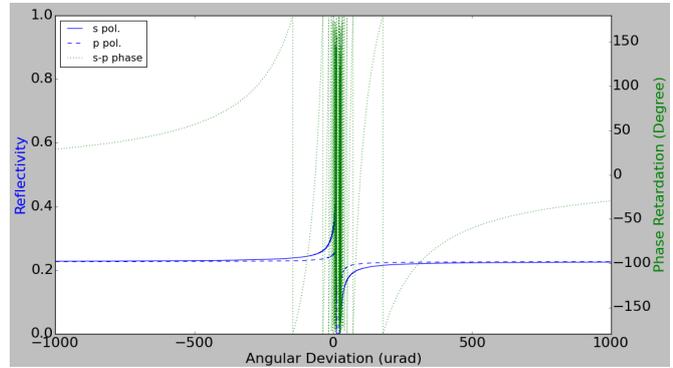


(b) CrystalViewer plot

Figure 5.3: Calculated transmittivities for  $\sigma$  (s) and  $\pi$  (p) polarizations and phase retardation (s - p) for diamond 220 Bragg transmission at 10 keV from a crystal 1-*mm* thick.



(a) 200 points



(b) 20000 points

Figure 5.4: The figure shows how the oscillation of the phase increases with the number of sample points inside the divergence region. The `numpy.unwrap` function can normally be used to eliminate these wrapping artifacts, but the presence of a divergent behavior makes it very difficult to properly unwrap the plot, since this leads to the rightmost values changing unpredictably with the number of points in the plot (i.e. the number of phase jumps).

## Si 220 Laue diffraction at 10 keV

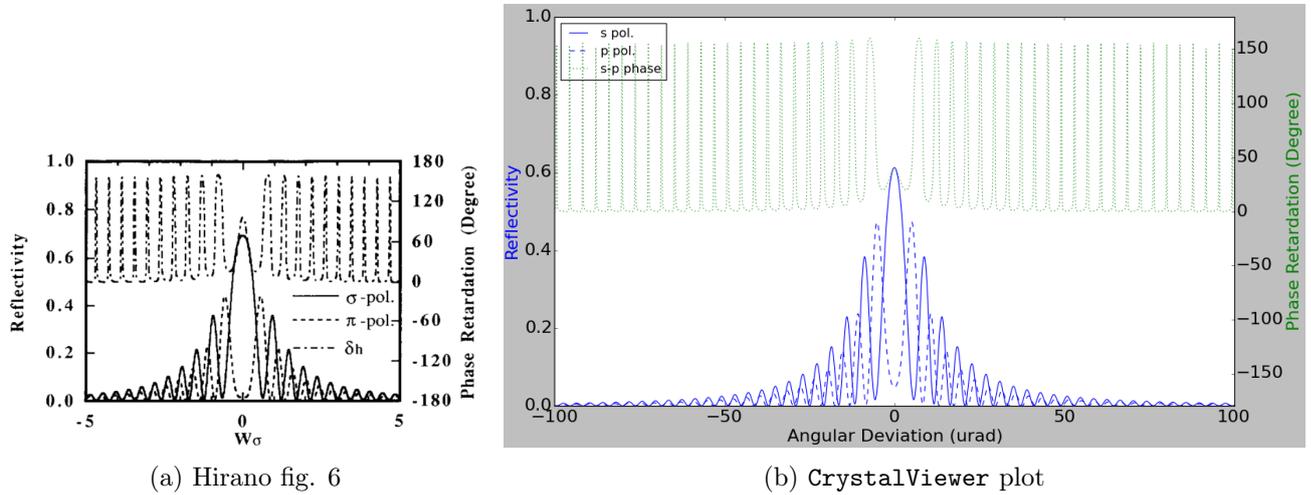


Figure 5.5: Calculated reflectivities for  $\sigma$  (s) and  $\pi$  (p) polarizations and phase retardation (s - p) for Si 220 Laue diffraction at 10 keV from a crystal  $50\text{-}\mu\text{m}$  thick.

## Ge 220 Laue diffraction at 20 keV

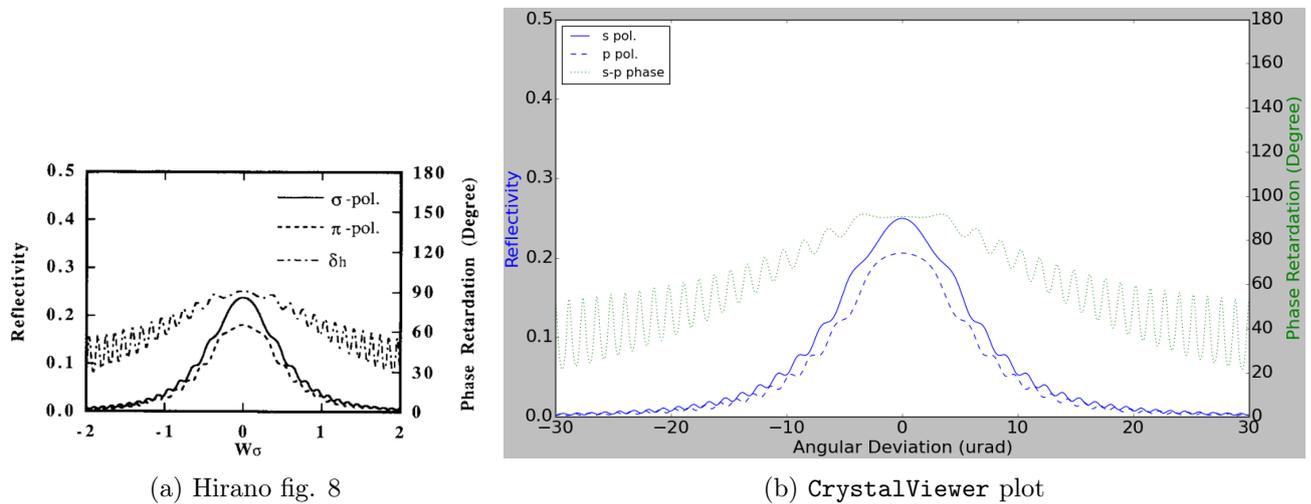
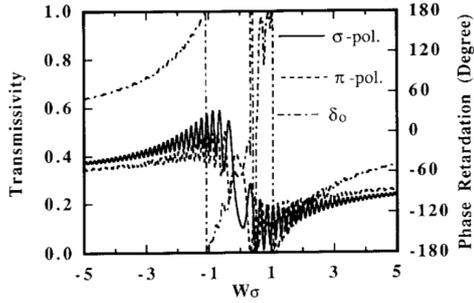
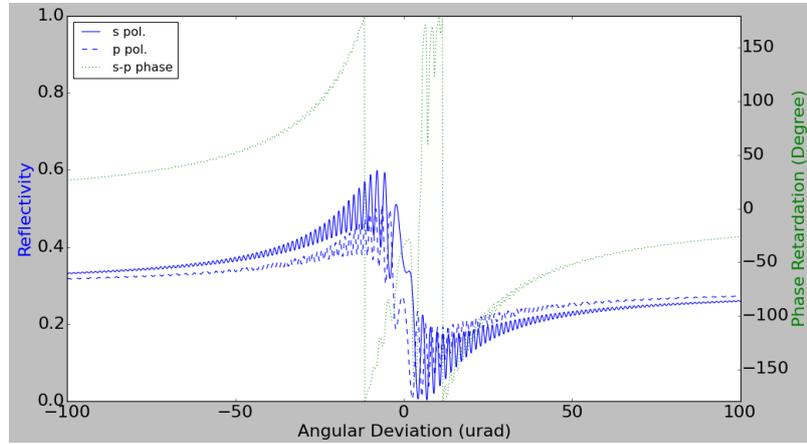


Figure 5.6: Calculated reflectivities for  $\sigma$  (s) and  $\pi$  (p) polarizations and phase retardation (s - p) for Ge 220 Laue diffraction at 20 keV from a crystal  $180\text{-}\mu\text{m}$  thick.

## Si 220 Laue transmission at 10 keV



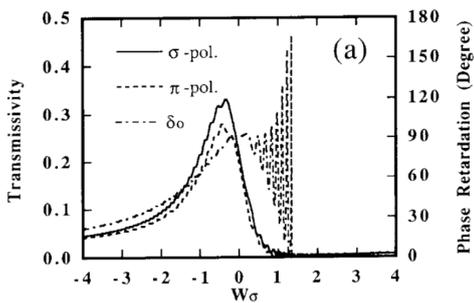
(a) Hirano fig. 9



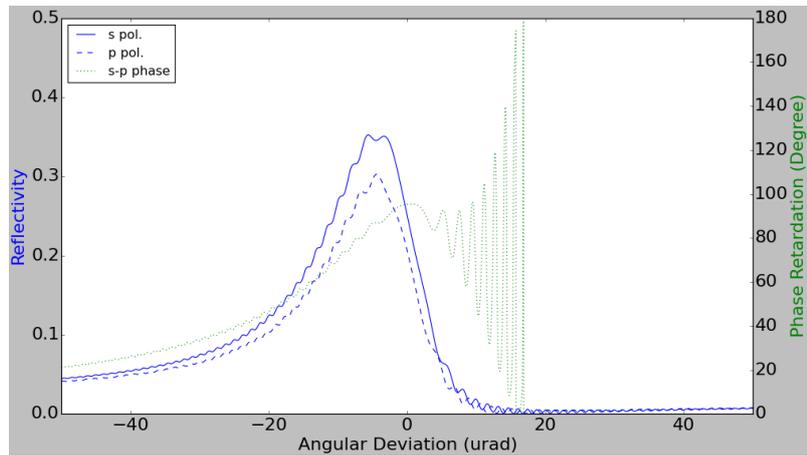
(b) CrystalViewer plot

Figure 5.7: Calculated transmittivities for  $\sigma$  (s) and  $\pi$  (p) polarizations and phase retardation (s - p) for Si 220 Laue transmission at 10 keV from a crystal  $150\text{-}\mu\text{m}$  thick.

## Ge 220 Laue transmission at 20 keV



(a) Hirano fig. 11a



(b) CrystalViewer plot

Figure 5.8: Calculated transmittivities for  $\sigma$  (s) and  $\pi$  (p) polarizations and phase retardation (s - p) for Ge 220 Laue transmission at 20 keV from a crystal  $180\text{-}\mu\text{m}$  thick.

The integration into the OASYS environment allows to simulate more complex systems with two or more crystals in succession. Figures 5.9 and 5.10 show Bragg-Laue optics for production of Circularly Polarized X-rays (CPX) as presented by Hirano *et al.* and a corresponding simulation using *crystalpy*. The *degree of circular polarization* is defined as

$$P_c = \frac{s_3}{s_0}. \quad (5.1)$$

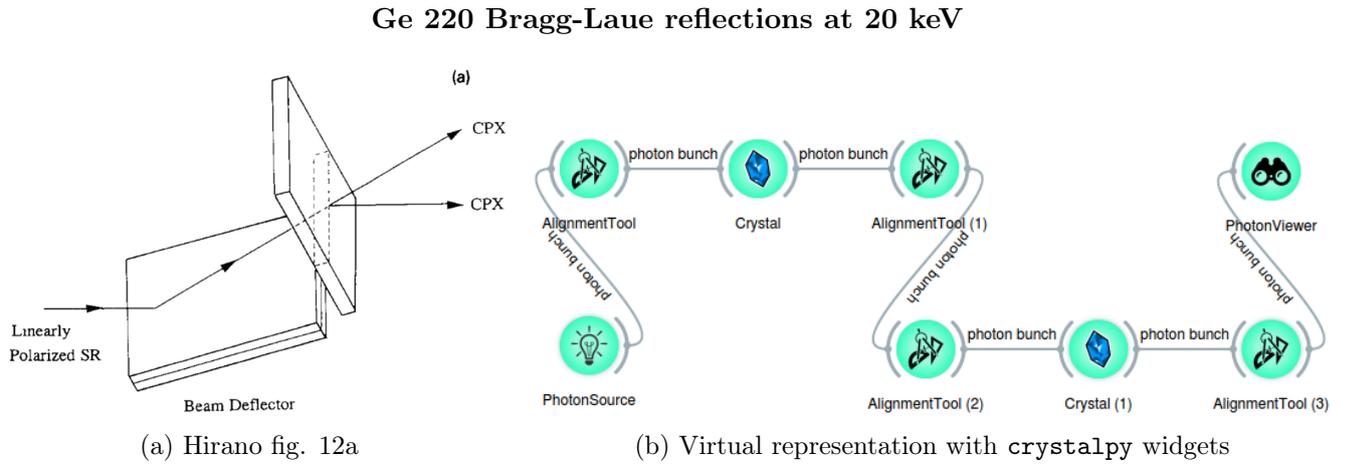


Figure 5.9: Bragg-Laue optics to produce CPX.

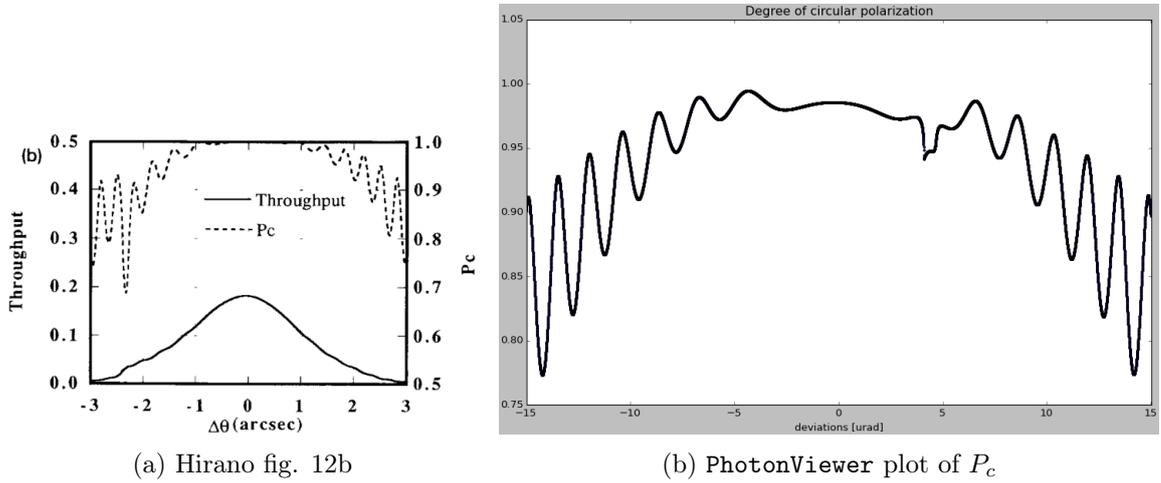


Figure 5.10: The photon beam is initialized with a Stokes vector  $(1 \ 0 \ 1 \ 0)$  and it undergoes Bragg and Laue diffraction from Ge 220 planes at 20 keV. The first crystal acts as a collimator, the second acts as phase retarder and is  $180 \mu\text{m}$  thick. The scattering plane is inclined by an angle  $\alpha = 48.25^\circ$ .

## 5.2 Testing against SHADOW

On top of the testing against results in published literature, tests were performed against software such as *polarsim* and SHADOW. The latter doesn't use Müller-Stokes calculus to compute the Stokes parameters, instead it keeps track of the complex amplitudes of the "photon" field components<sup>1</sup> during the interaction with the beamline's optical elements. Amplitudes and phases can then be used to obtain the Stokes vector according to 1.11.

Since *crystalpy* and SHADOW follow different procedures to get to the same result, a correspondence between the two is a sign things are most probably working as expected in both programs, as a systematic error is not likely.

The  $s_1$  and  $s_3$  parameters have opposite signs in the SHADOW and *crystalpy* simulations because in SHADOW  $s_1 = |E_\pi|^2 - |E_\sigma|^2$  and  $s_3 \propto \sin(\phi_\sigma - \phi_\pi)$ , whereas in *crystalpy*  $s_1 = |E_\sigma|^2 - |E_\pi|^2$  and  $s_3 \propto \sin(\phi_\pi - \phi_\sigma)$  4.2.

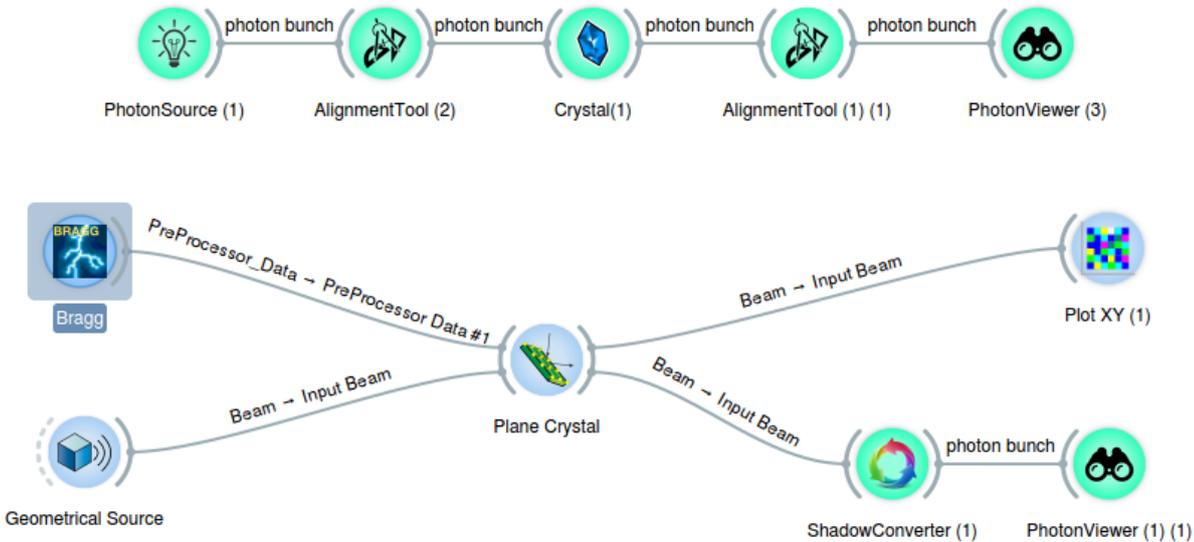


Figure 5.11: The upper part of the figure shows the optical setup for crystal diffraction consisting of the familiar *crystalpy* widgets.

The lower part shows an equivalent setup made of SHADOW widgets: Bragg preprocessor creates a file with data about the crystal structure, Geometrical Source emits photons in random directions from a custom area (a point, a rectangle, an ellipse, ...), Plane Crystal performs the diffraction computations, and PlotXY can plot several parameters including intensity and phases for the different polarization components and the Stokes parameters.

<sup>1</sup>Note that the term photon is not used with its physical meaning, but to refer to a `Photon` object.

## Si 111 Bragg diffraction at 10 keV

$$s_{initial} = (1 \ 0 \ 1 \ 0)$$

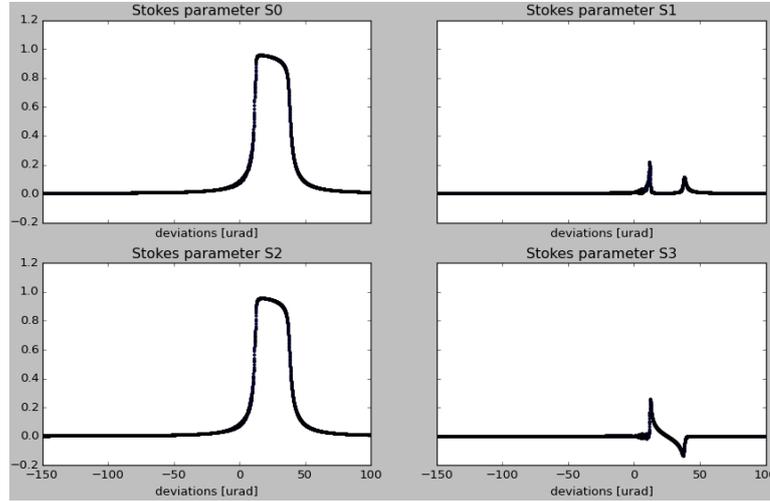


Figure 5.12: Stokes parameters for L+45°-polarized x-rays that underwent Si 111 Bragg diffraction at 10 keV from a 100- $\mu\text{m}$  crystal with  $\alpha = 0$  calculated with `crystalpy` widgets.

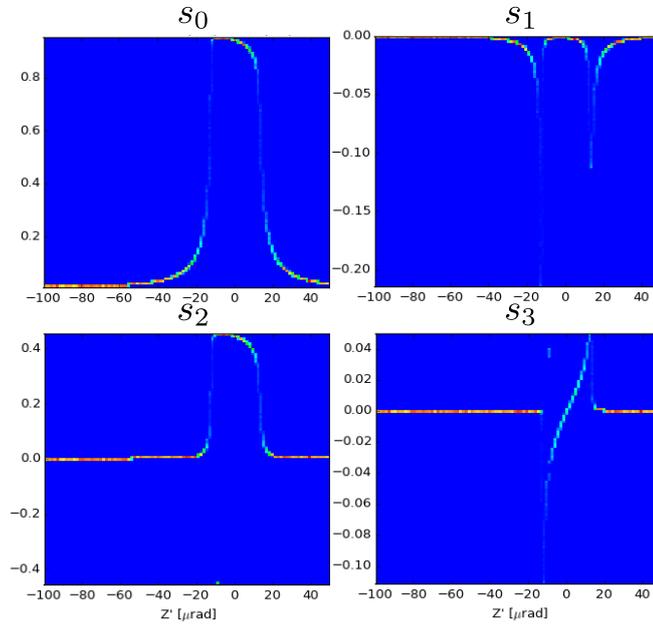


Figure 5.13: The same calculation performed with `SHADOW` widgets. The fact that  $s_2$  and  $s_3$  are half the `crystalpy` values can be attributed to a difference in the definition of the field amplitudes:  $E_0^{SHADOW} = E_0^{crystalpy} / \sqrt{2}$ .

As a last test, let us examine whether the Stokes parameters computed by the `BendingMagnet` widget match those from SHADOW's bending magnet.

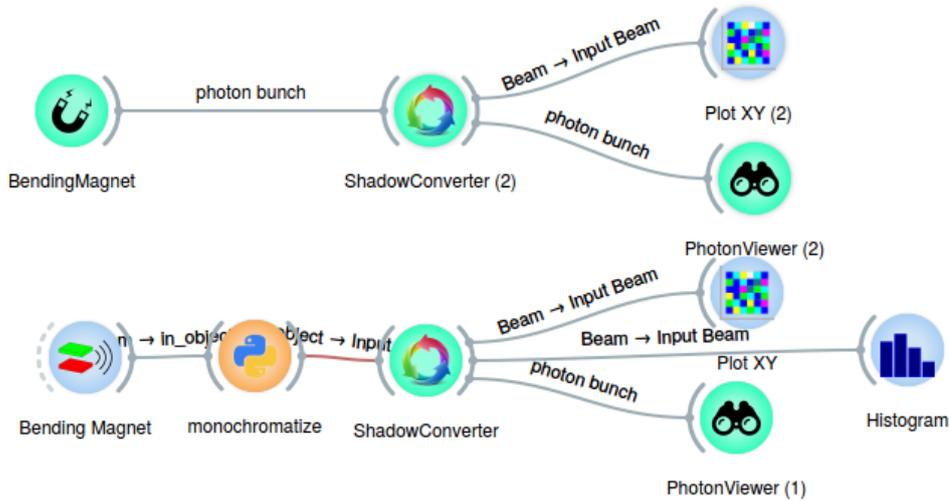


Figure 5.14: At the top: setup with `crystalpy`'s `BendingMagnet` widget and different viewers. At the bottom: setup with SHADOW's bending magnet.

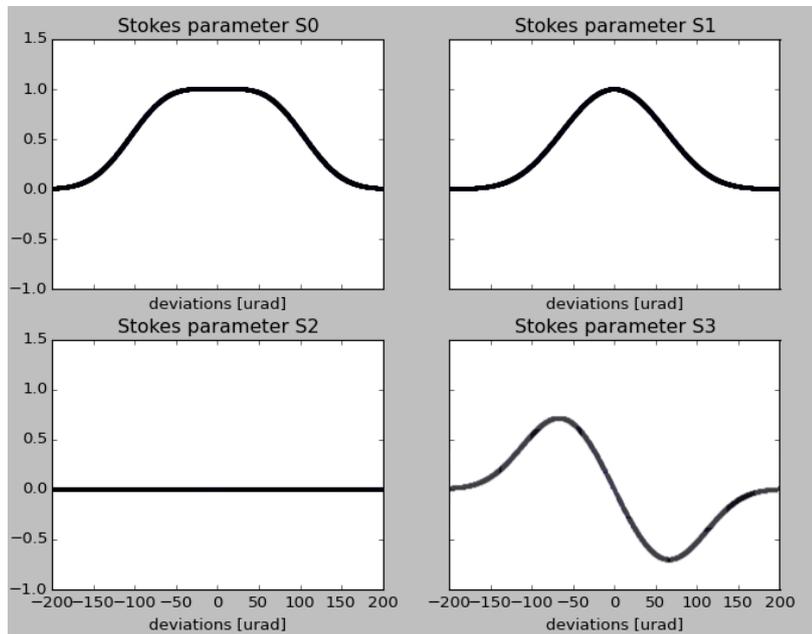


Figure 5.15: Stokes parameters computed by `crystalpy`'s `BendingMagnet` widget. The parameters are observed for the radiation at  $8keV$  generated with electron beam energy  $E = 6GeV$ , electron beam current  $I = 0.2A$ , horizontal divergence  $1mrad$  and magnetic radius  $R = 25m$ .

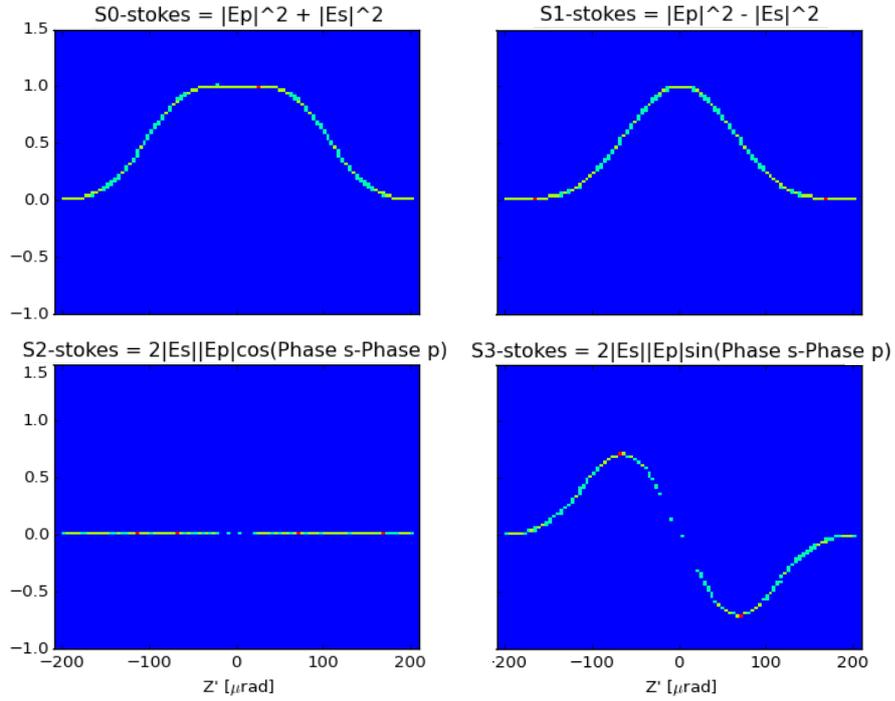


Figure 5.16: Stokes parameters computed by SHADOW's Bending Magnet widget. The parameters are observed for the radiation at  $8\text{keV}$  generated with electron beam energy  $E = 6\text{GeV}$ , electron beam current  $I = 0.2\text{A}$ , horizontal divergence  $1\text{mrad}$  and magnetic radius  $R = 25\text{m}$ .

The comparison with the results by Hirano *et al.* and with SHADOW show nearly perfect agreement. This proves that both the **diffraction** and **polarization** toolboxes work as expected. Nonetheless, more tests are needed to make sure the tool is reliable: testing against data from [8, 9] and [39] could be the next steps, as well as starting to use **crystalpy** alongside SHADOW for real beamline simulations at ESRF. Some more effort also needs to be put into the integration into OASYS, especially when linking together multiple crystals.

# Conclusions

In this thesis I have presented the formalism of polarization states, the theory of dynamical diffraction of X-rays from perfect crystals and how I used both to create a Python library which is able to simulate the changes in X-ray polarization brought about by crystal diffraction. I also showed how I integrated the `crystalpy` library into the OASYS Graphical Environment in the form of user-friendly widgets that can be included in a complete beamline simulation.

The calculations performed with `crystalpy` match those found in existing literature, in particular they are in good agreement with the results obtained by Hirano *et al.* [4]. Moreover, the comparison of the results with corresponding SHADOW simulations allowed to eliminate some bugs in the latter.

The code is still under development and needs to be further tested. It will be improved in the coming months to work better with multiple crystals and it will be applied to real-world cases in collaboration with ESRF beamlines which employ crystal phase plates (e.g. BM28). In the future it will be used as cross-reference for crystal simulations in many OASYS packages, and will eventually replace and renew the crystal code in ShadowOui.



# Appendix A.

## Bending Magnet Radiation

The `BendingMagnet` widget in the `oasys-crystalpy` library simulates the radiation from a bending magnet in the Stokes formalism. It calculates the polarization properties of the radiation emitted by a relativistic charged particle on a circular path, following the treatments by Sokolov and Ternov [37] and Green [40].

A bending magnet is a curved electromagnet used to bend the electrons into their racetrack orbit. As the electrons are deflected from their straight path when passing through the magnet, they emit X-rays tangentially to the plane of the electron beam like a "sweeping searchlight". The synchrotron light from a bending magnet covers a wide and continuous spectrum, from microwaves to hard X-rays [41].

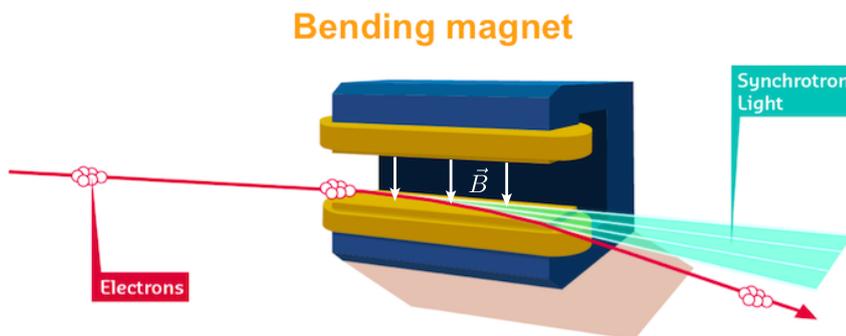


Figure A.1: Schematic representation of a bending magnet (figure from [41]).

The power radiated by a *non-relativistic* accelerated charge is given by Larmor's formula [13] and is distributed in a torus around the centripetal acceleration vector. For *ultrarelativistic* electrons<sup>2</sup> however, Larmor's formula must be generalized to be invariant under Lorentz transformation. The relativistic formula predicts that the power be distributed in a narrow cone tangential to the instantaneous trajectory as shown in figure A.2.

<sup>2</sup>An electron moving at near the speed of light, i.e. for which relativistic effects are prominent ( $\gamma \gg 1$ )

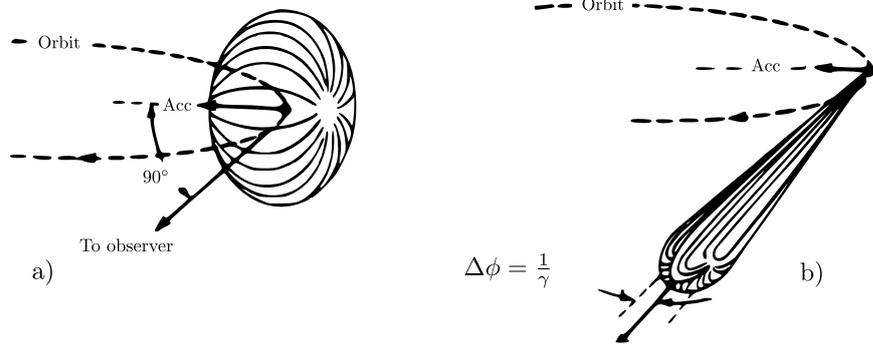


Figure A.2: Figure (a) shows the Larmor emission profile for a non-relativistic charged particle on a circular trajectory. Figure (b) shows the ultrarelativistic emission profile for the same particle when  $v \approx c$ .

Synchrotron radiation is normally generated over an orbital arc  $\theta$  much larger than the radiation angle of emission, and is usually collected in such a manner as to sum over the angles in the orbital plane [40]. Let us compute the power emitted by an electron beam with energy  $E$ , current  $I$ , critical energy  $\epsilon^3$ , and horizontal divergence  $\theta$ . The power associated with radiation of energy  $\epsilon$  and observed at an angle  $\psi$  from the orbit plane can be calculated using the formula [40, p. 6]:

$$\frac{dP}{d\psi} = f(E(\text{GeV}), I(\text{A}), \epsilon_c(\text{eV}), \theta(\text{mrad})) \times \left\{ (1 + \gamma^2 \psi^2)^{3/2} \left[ l_2 K_{2/3}(\xi) + l_3 \gamma \psi (1 + \gamma^2 \psi^2)^{-1/2} K_{1/3}(\xi) \right] \right\}^2 \quad (5.2)$$

where

$$\xi = \frac{\epsilon}{2\epsilon_c} (1 + \gamma^2 \psi^2)^{3/2},$$

$\gamma$  is the *Lorentz factor*,  $K_{m/n}$  are the *Bessel functions* of the third kind,  $l_2$  and  $l_3$  are the parameters which describe the polarization state. In particular,

$$\begin{aligned} \text{Linear Horizontally Polarized (LHP)} &\implies l_2 = 1, \quad l_3 = 0; \\ \text{Linear Vertically Polarized (LVP)} &\implies l_2 = 0, \quad l_3 = 1; \\ \text{Right Circularly Polarized (RCP)} &\implies l_2 = 1/\sqrt{2}, \quad l_3 = -l_2; \\ \text{Left Circularly Polarized (LCP)} &\implies l_2 = 1/\sqrt{2}, \quad l_3 = l_2. \end{aligned}$$

The phase difference  $\delta$  between the horizontal and vertical field components  $E_x$  and  $E_y$  is given by Sokolov (3.23):

$$\delta = \delta_y - \delta_x = \sin^{-1} \frac{P_{\text{LCP}} - P_{\text{RCP}}}{2\sqrt{P_{\text{LHP}}P_{\text{LVP}}}} \quad (5.3)$$

<sup>3</sup>The *critical energy* divides the emission spectrum in two parts, each containing half of the total radiated power. It can be calculated as  $\epsilon_c(\text{keV}) = 2.2183E(\text{GeV})^3/R(\text{m})$ , where  $R$  is the *Larmor radius* of an electron in a magnetic field  $B$  [40, p. 3].

From these elements we can calculate the Stokes parameters for bending magnet radiation:

$$\left. \begin{aligned} s_0 &= P_{\text{LHP}} + P_{\text{LVP}}, \\ s_1 &= P_{\text{LHP}} - P_{\text{LVP}}, \\ s_2 &= 2\sqrt{P_{\text{LHP}}}\sqrt{P_{\text{LVP}}}\cos\delta, \\ s_3 &= 2\sqrt{P_{\text{LHP}}}\sqrt{P_{\text{LVP}}}\sin\delta. \end{aligned} \right\} \quad (5.4)$$

The `BendingMagnet` widget uses these equations to create a `PolarizedPhotonBunch` object made up by photons with different angular deviations  $\psi$  from the horizontal direction in the orbit plane (conventionally the y-axis).

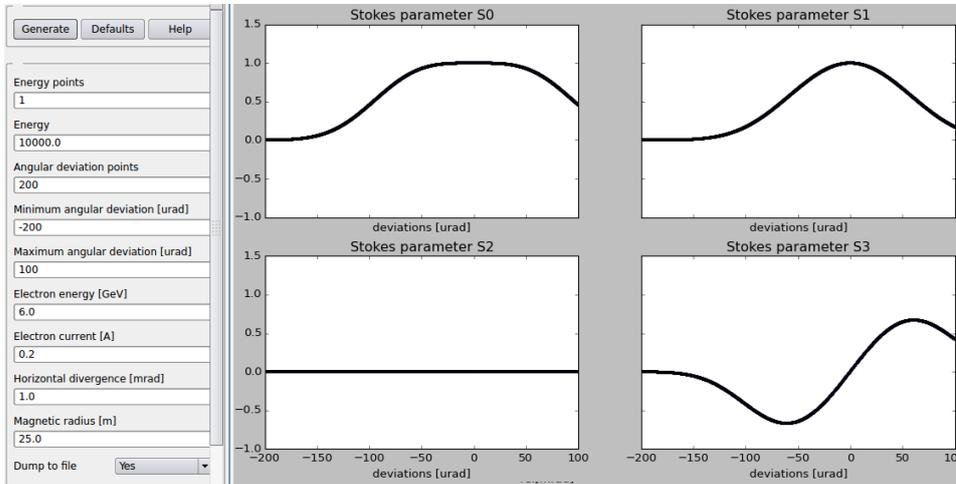


Figure A.3: Example of Stokes parameters for different angular deviations from the orbit plane calculated with the `BendingMagnet` widget. On the left the user can set the necessary parameters. The default values correspond to the ESRF beam.



# Bibliography

- [1] F. de Bergevin and M. Brunel. “Observation of Magnetic Superlattice Peaks by X-ray Diffraction on an Antiferromagnetic NiO Crystal”. In: *Physics Letters* 39A.2 (1972).
- [2] M. Blume and Doon Gibbs. “Polarization dependence of magnetic x-ray scattering”. In: *Physical Review B* 37.4 (1988).
- [3] J. P. Hill and D. F. McMorrow. “X-ray Resonant Exchange Scattering: Polarization Dependence and Correlation Functions”. In: *Acta Crystallographica A* 52 (1996).
- [4] Keiichi Hirano, Tetsuya Ishikawa, and Seishi Kikuta. “Perfect crystal X-ray phase retarders”. In: *Nuclear Instruments and Methods in Physics Research* 336 (Nov. 1993).
- [5] Kwang-Je Kim. “Optical and power characteristics of synchrotron radiation sources”. In: *Optical Engineering* 34 (1995).
- [6] David Attwood. *Soft X-rays and Extreme Ultraviolet Radiation: Principles and Applications*. 1st ed. Cambridge University Press, 1999.
- [7] P. Elleaume. “Undulators and wigglers for the new generation of synchrotron sources”. In: *Journal of Synchrotron Radiation* 1 (1994).
- [8] Laurence Bouchenoire et al. “Performance of phase plates on the XMaS beamline at the ESRF”. In: *Journal of Synchrotron Radiation* 10 (2003).
- [9] Laurence Bouchenoire et al. “Polarisation optimisation for ferromagnetic diffraction: A case study of iron”. In: *Nuclear Instruments and Methods in Physics Research A* 566 (2006).
- [10] William H. Zachariasen. *Theory of X-Ray Diffraction in Crystals*. 1st ed. Dover Publications, 1967.
- [11] Manuel Sanchez del Rio and Luca Rebuffi. “A proposal for an Open Source graphical environment for simulating X-ray optics”. In: *proceedings of SPIE* 9209 (2014).
- [12] Manuel Sanchez del Rio et al. “SHADOW3: a new version of the synchrotron X-ray optics modelling package”. In: *Journal of Synchrotron Radiation* (2011).

- [13] John D. Jackson. *Classical Electrodynamics*. 2nd ed. John Wiley & Sons, 1975.
- [14] Max Born and Emil Wolf. *Principles of Optics*. 6th ed. Pergamon Press, 1980.
- [15] Dennis H. Goldstein. *Polarized Light*. 3rd ed. CRC press, 2011.
- [16] Hiroyuki Fujiwara. *Spectroscopic Ellipsometry: Principles and Applications*. John Wiley & Sons, 2007.
- [17] Serge Huard. *Polarisation de la lumière*. Masson, 1994.
- [18] Neil W. Ashcroft and N. David Mermin. *Solid State Physics*. 1st ed. Saunders College Publishing, 1976.
- [19] Jens Als-Nielsen and Des McMorrow. *Elements of Modern X-ray Physics*. 2nd ed. John Wiley & Sons, 2011.
- [20] Boris W. Batterman and Henderson Cole. “Dynamical Diffraction of X Rays by Perfect Crystals”. In: *Reviews of Modern Physics* 36.3 (July 1964). URL: <http://journals.aps.org/rmp/abstract/10.1103/RevModPhys.36.681>.
- [21] *ESRF Upgrade Programme*. URL: <http://www.esrf.eu/about/upgrade>.
- [22] *The Purple Book*. 2007. URL: [http://www.esrf.eu/Archive\\_files/Upgrade/ESRF-SciTechProg2008-2017.pdf](http://www.esrf.eu/Archive_files/Upgrade/ESRF-SciTechProg2008-2017.pdf).
- [23] *The Orange Book*. 2014. URL: [http://www.esrf.eu/Archive\\_files/Upgrade/ESRF-orange-book.pdf](http://www.esrf.eu/Archive_files/Upgrade/ESRF-orange-book.pdf).
- [24] Manuel Sanchez del Rio. “Simulations of X-ray optics for ESRF-Upgrade Programme”. In: *proceedings of SPIE* 9209 (2014).
- [25] Oleg Chubar and Pascal Belleaume. “Accurate and Efficient Computation of Synchrotron Radiation in the Near Field Region”. In: *proceedings of the EPAC98 Conference* (1998), pp. 1177–1179.
- [26] Xianbo Shi et al. “A hybrid method for X-ray optics simulation: combining geometric ray-tracing and wavefront propagation”. In: *Journal of Synchrotron Radiation* (2014).
- [27] E. Bergback Knudsen et al. “McXtrace: a Monte Carlo software package for simulating x-ray optics, beamlines and experiments”. In: *Journal of Applied Crystallography* 46 (2013).
- [28] Konstantin Klementiev and Roman Chernikov. “Powerful scriptable ray tracing package xrt”. In: *proceedings of SPIE* (2014).
- [29] *OASYS*. URL: <http://ftp.esrf.eu/pub/scisoft/0asys/readme.html>.

- [30] Janez Demšar, Tomaž Curk, Aleš Erjavec, et al. “Orange: Data Mining Toolbox in Python”. In: *Journal of Machine Learning Research* 14 (2013).
- [31] *SOS Workshop*. 2016. URL: <https://www.elettra.eu/Conferences/2016/SOS/Main/HomePage>.
- [32] Carlos Giles et al. “Energy-dispersive phase plate for magnetic circular dichroism experiments in the X-ray range”. In: *Journal of Applied Crystallography* 27 (1994).
- [33] Carlos Giles and Christian Vettier. “X-ray polarimetry with phase plates”. In: *Review of Scientific Instruments* 66 (1998).
- [34] Olinde Rodrigues. “Des lois géométriques qui regissent les déplacements d’ un système solide dans l’ espace, et de la variation des coordonnées provenant de ces déplacements considérées indépendent des causes qui peuvent les produire”. In: *Journal de Mathématiques Pures et Appliquées* 5 (1840).
- [35] Manuel Sanchez del Rio et al. “Simulation of X-ray diffraction profiles for bent anisotropic crystals”. In: *Journal of Applied Crystallography* 48 (2015).
- [36] Tom Schoonjans, Manuel Sanchez del Rio, et al. “The xraylib library for X-ray–matter interactions. Recent developments”. In: *Spectrochimica Acta Part B* 66 (2011).
- [37] Arseny A. Sokolov and Igor M. Ternov. *Synchrotron Radiation*. 1st ed. Pergamon Press, 1968.
- [38] *matplotlib*. URL: <http://matplotlib.org/>.
- [39] Carlos Giles. “Étude de lames quart d’onde pour les rayons X et application à un montage dispersif en énergie”. Titre de Docteur en Sciences. Université de Paris 7, 1995.
- [40] George K. Green. *Spectra and Optics of Synchrotron Radiation*. Brookhaven National Laboratories, 1976.
- [41] *ESRF Website*. URL: <http://www.esrf.eu/about/synchrotron-science/synchrotron>.
- [42] *Orange*. URL: <https://github.com/biolab/orange3>.
- [43] William A. Shurcliff. *Polarized Light: Production and Use*. Harvard University Press, 1962.
- [44] Igor M. Ternov. “Synchrotron radiation”. In: *Uspekhi Fizicheskikh Nauk (Advances of Physical Sciences)* 38 (1995).
- [45] Manuel Sanchez del Rio and Roger J. Dejus. “XOP v2.4: recent developments of the X-ray optics software toolkit”. In: *proceedings of SPIE* (2011).